

全国高等学校 夏の課題解決 協働探究コンテスト 2025

Unityゲームでつくる!  
すいかばマンと  
暑い夏に  
立ち向かおう

BIG  
スライカバー

ロッテ  
種類別氷菓 スライカ果汁5%



応募はこちら!



- 【審査員】
- すいかばマン (本人)
  - 鹿野利壽様 (一般社団法人デジタル人材共創連盟)
  - 丸子宏幸様 (株式会社ロッテマーケティング本部 第一ブランド戦略部アイス企画課)
  - 金澤直樹様 (株式会社ロッテ 経営戦略部事業開発課)
  - 加本満里奈様 (株式会社MIXIモンスターIPクリエイション部)
  - 高井志保様 (一般社団法人Women AI Initiative Japan)

大会会場  
SUMIDA INNOVATION CORE

学校で  
出張授業も  
OK!

予選会 (書類審査)

敗者復活戦・1DAYイベント

本戦 ※本戦はリモート参加可能

2025年6月3日 (火)  
~2025年7月18日 (金)

2025年7月26日 (土)

2025年7月27日 (日)  
時間 12:30~16:00

主催：株式会社ロッテ  
共催：一般社団法人デジタル共創連盟  
事務局：株式会社スクーミー



MADE IN YAMANASHI  
初版発行 2025年7月16日 (水)



4 573676 920164  
定価：本体 1,650 円  
(税込 1,500 円税 10%)



Unityでゲームをつくらう  
すいかばマン協働探究コラボver

発行 株式会社スクーミー 2025年発行 Unityでゲームをつくらう  
〒4000015 山梨県甲府市大手1-2-21 イノベーション・ソサエティ101



# 月刊みんなの タビスタ

特別企画

## Unityでゲーム をつくらう

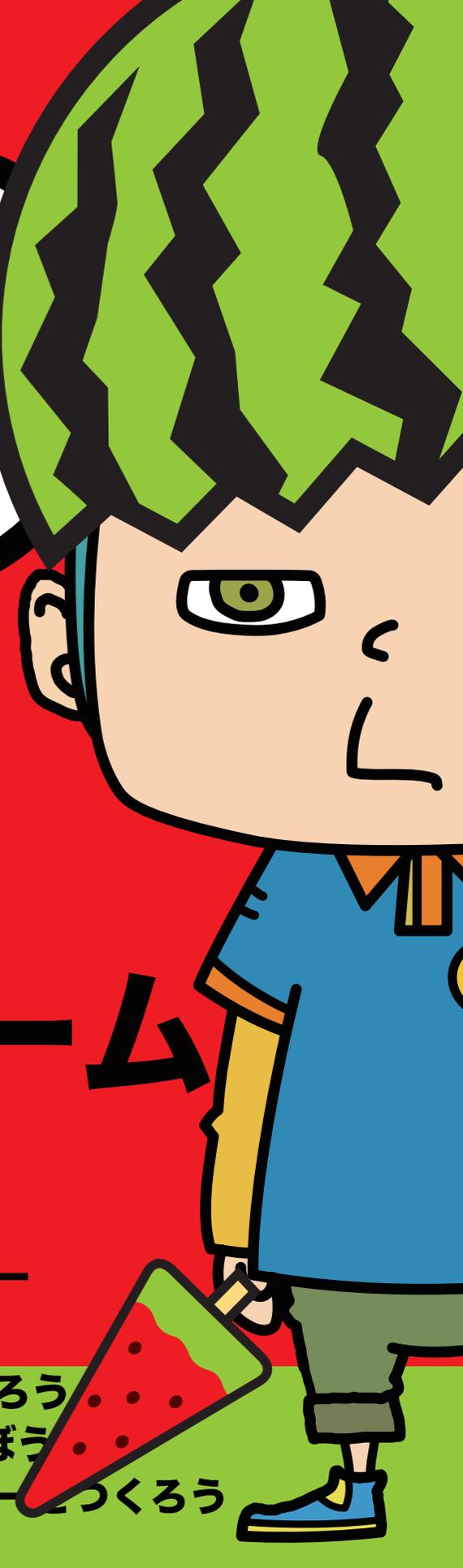
ピックアップセンサー 角度センサーコネクター

すいかバーとメディアミックス

すいかばマンデザインの使い方について知ろう

情報システムをUnityとスクーミーで学ぼう

センサーコネクターでゲームコントローラーをつくらう



ムサシは スクーミーワールド  
という世界にいる小学3年生の男の子。

# あやしい。

・・・と、ムサシは

いつもいろんなことをあやしんでいる。

そんなムサシには ナカマがいる。

じぶんのトクイを いかしているヒトたちばかりだ。

ムサシが あやしんでいることは  
アネのリツコが みんなに しっかりホウコクしている。  
だから みんなも しっかり知っている。

あるとき スイカバーをかってもらったムサシは  
おかしうりばにいるアネのリツコに  
みて！すいかばマンがグミにもガムにいるよ！  
といわれたそうです。

どうしてアイスうりばだけではなく  
おかしコーナーにも すいかばマンがいるんだろうと  
あやしんでいるムサシをみて  
リツコは きょうもみんなにホウコクをします。

「ムサシがすいかばマンをみてあやしんでました！」と。



# 情報社会の変化を スイカパーの歴史から考えよう

## 1 情報技術の発達と社会変化

情報技術の発達は、社会や人々の生活に大きな影響を与えています。企業活動においても、情報の収集・処理・発信の方法が大きく変化してきました。情報技術の進歩により、企業のマーケティング手法は劇的に変化しています。従来の一方向的な情報発信から、顧客との双方向コミュニケーションが可能になりました。

**昔** 1986年のスイカパー発売当初のチラシ  
「くだもの屋さんもビックリ!!」

ロッテのスイカパーは1986年の発売当初、紙のチラシによる商品案内が主流でした。「くだもの屋さんもビックリ!!」というキャッチコピーが印刷された一方向的な情報発信でした。



**今** 現在のスイカパー公式サイト  
「ロッテランド」

現在では、デジタル技術を活用したWebサイトでの情報発信や、キャラクターを通じた親しみやすいコミュニケーションが実現されています。38年間の変化は、情報技術の発達が企業活動に与えた影響を如実に示しています。



すいかばマン

## すいかばマンと考えよう!

もしあなたが商品の宣伝担当者だったら…?  
1986年と現在で、**どのような違いがありますか?**  
それぞれの時代の良い点・課題点をグループで話し合ってみましょう。  
また、**10年後にはどのような新しい宣伝方法が生まれると思いますか?**



## 2 情報通信技術の発達と新しいコミュニケーション

情報通信技術の発達により、新しいコミュニケーション手段が生まれてきました。SNS、動画配信、インタラクティブなWebサイトなど、多様なメディアが登場しています。コミュニケーション手段にはそれぞれ特徴があり、相手との関係などを考慮して、適切な手段を選択する必要があります。また広告や宣伝では、複数のメディアを活用したメディアミックスやクロスメディアといった考え方が重要となります。各メディアの特性を活かし、相乗効果を狙った統合的なアプローチが求められます。



▲スイカパーでは、基本商品（BIG スイカパー）から多様な展開（スイカ & メロンパー、限定商品）により、異なる顧客層へのアプローチを図っています。「2023年ピンクに染まった!? スイカパー」「2022年幸せ!?!の青いスイカパー」「2019年謎のスイカパー」など、話題性のある商品展開で顧客の関心を引く戦略です。

▲スイカパーでは、物理的なパッケージからデジタルコンテンツへの導線設計が効果的に行われています。パッケージのQRコードをスキャンすることで、ロッテランドの公式サイトにアクセスし、より詳細な情報を得ることができます。

▶SNSで関係で  
つくることができそう



すいかばマン

かばメロンちゃん

## すいかばマンと考えよう!

- 「誰にでも使いやすいデザイン」を考える時、**どのような工夫が必要でしょうか?**  
スイカパーのパッケージを例に、年齢、文化、障がいの有無などを考慮したデザインのアイデアをグループで考えてみましょう。**QRコード以外にも、どのような方法で情報を伝えることができるでしょうか?**
- なぜスイカパーは毎年違う色や味の限定商品を出すのでしょうか?**この戦略にはどのような効果がある**と思いますか?  
また、皆さんが新しいスイカパーを企画するとしたら、**どのような商品を作りますか?**ネーミングも含めて考えてみましょう。



すいかばマン

より良い開発を進めるために!

# プロジェクト・マネジメントと コンテンツをすいかばマンたちと考えよう

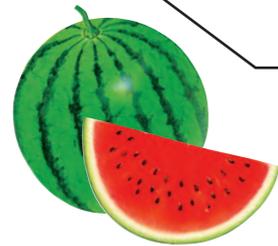
## 1 情報セキュリティと知的財産権



企業などの組織では、個人の情報モラルに任せるだけでなく、情報の取り扱い方法や手順をまとめた情報セキュリティポリシーを定め、対策をとっています。  
知的財産権の保護は、情報社会における重要な課題の一つです。企業は自社の無形資産を適切に管理し、法的保護を受ける必要があります。  
また情報があふれる情報社会では、受け手に分かりやすく的確に情報を伝達するために情報デザインが必要となります。情報デザインによって、情報が正確に、役に立つ形で伝わり、人や社会に影響を与えることができます。

例えば

ロッテは2018年7月27日に、キャラクター「すいかばマン」(登録番号第6065472号)と「かばメロちゃん」(登録番号第6065471号、第6065473号)の商標登録を行いました。これは企業が情報セキュリティポリシーの一環として知的財産を保護している実例です。



すいかば君



すいかばちゃん



すいかば君



すいかば美ちゃん

例えば

スイカバ-のキャラクターデザインは時代とともに変化しています。発売当時の「すいかば君」「すいかばちゃん」「すいかば美ちゃん」から、現在の「すいかばマン」「かばメロちゃん」へと進化し、より現代的で親しみやすい表現になっています。

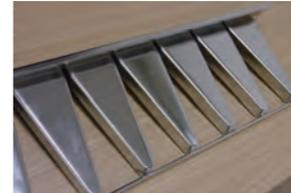


より良い開発を進めるために!

# プロジェクト・マネジメントと デザインをすいかばマンたちと考えよう

## 2 情報セキュリティと知的財産権

▶ スイカバ-の独特な三角形は、「どうすればスイカであることが伝わるかを考えた結果、スイカを家庭で食べる時の形状がイメージとして伝わりやすい」という理由で採用されました。この特徴的な形状を活かした撮影技法により、商品の認知度向上を図っています。



写真を撮影する際に、被写体に対する角度や構図を変えることで、伝えたいことを強調して表現することができます。視覚的なインパクトは、情報伝達において重要な要素です。動画の魅力では、角度や構図による表現の仕方は静止画と共通します。しかし、カメラの動きや被写体の動きなど、文字や静止画では伝えられない情報を伝えることが可能になります。コンテンツを制作する場合、伝えたい情報に応じて、文字や画像などをいかに組み合わせるかが重要となります。五感に訴える総合的な表現により、より効果的な情報伝達が可能になります

スイカバ-では、味覚への訴求も重要な要素として位置づけられています。  
チョコ種は「スイカバ-のためにチョコから開発した原料」として専用開発され、赤部分は「甘く赤いイメージ」、緑部分は「甘くておいしい緑色のイメージ」で味づくりが行われています。



### カラーガイド (コルカラー)

※このガイドはあくまでイメージの参考としてご利用ください。キャラクターのデザインは必ずご確認ください。



### カラーガイド (カラーパレット) 展開イメージ

カラーパレット展開イメージです。キャラクターの色のイメージを参考にしてください。カラーパレットを使用する場合は、必ずイメージを必ずご確認ください。



### キャラクター注意事項

下記のような使用法は禁止します。



### ホ-ジツク設定

すいかばマンとかばメロちゃんのホ-ジツク設定は可能です。撮影に際しては必ず、撮影のつもりは NG、カメラのフラッシュは必ずオフにしてください。ホ-ジツク設定は、必ずイメージを必ずご確認ください。



### すいかばマンの キャラクター設定

スイカバ-のキャラクターには詳細な設定が与えられており、動的な表現が可能です。  
すいかばマン：「暑い夏に立ち向かうヒーロー！」として正義感が強く、地球温暖化を気にする現代的設定  
かばメロちゃん：やさしくておしゃれ好きで、変身スーツのデザインを担当する創造的性格



かばメロちゃん

## すいかばマン と考えよう!

01 なぜ企業はキャラクターの商標登録をする必要があるのでしょうか？  
もし商標登録をしていなかったら、どのような問題が起こる可能性があるか、グループで考えてみましょう。

02 キャラクターのデザインが変化する理由は何でしょうか？時代に合わせてデザインを変えることの良い点と注意すべき点をグループで話し合ってみましょう。  
また、現在の「すいかばマン」「かばメロちゃん」のデザインには、どのような現代的要素が含まれていると思いますか？



すいかばマン



すいかばマン

## すいかばマン と考えよう!

すいかばマンが「地球温暖化を気にする」という設定には、どのような意味があるでしょうか？  
キャラクターに現代的な社会問題への関心を持たせることの効果を考えてみましょう。また、すいかばマンの友達を皆さんがオリジナルキャラクターとして作るとしたら、どのような性格や社会の関心事、テーマを設定しますか？



チョコかばマン  
暑が苦手な住んでいる  
すいかばマンのお友達！  
顔が(赤)くて暑さなんが  
へっちゃらさっ!



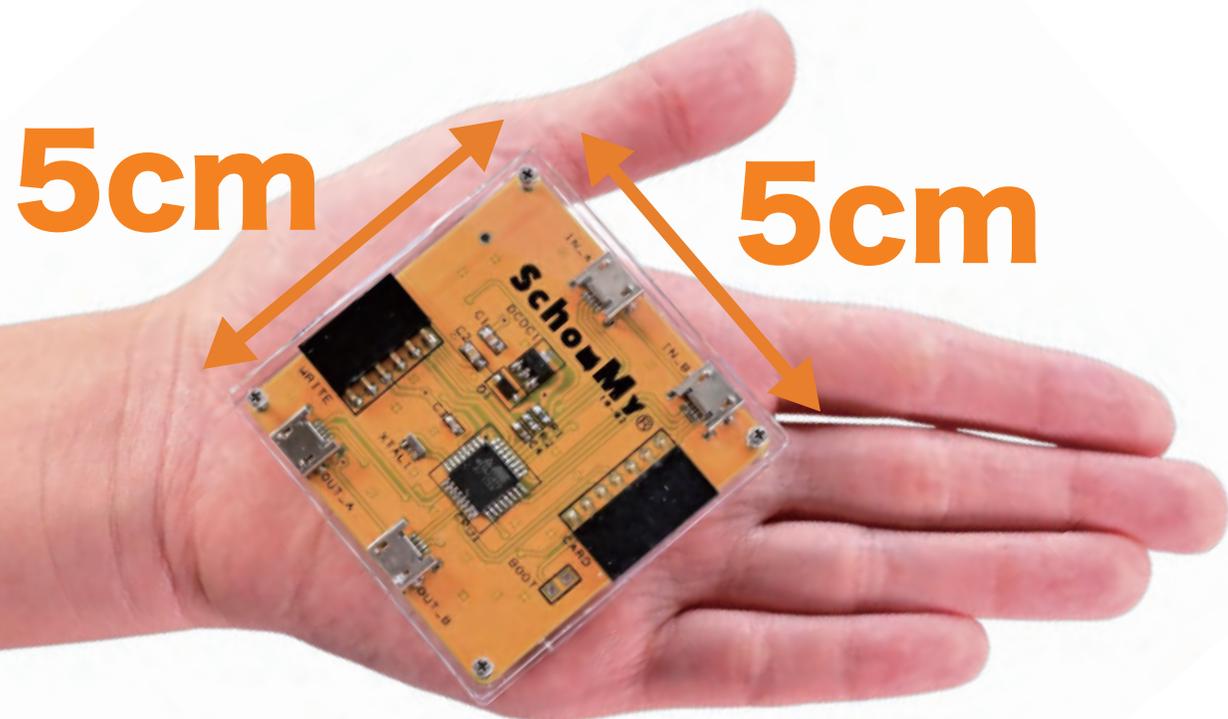
かばメロちゃん  
暑が苦手な住んでいる  
すいかばマンのお友達！  
顔が(赤)くて暑さなんが  
へっちゃらさっ!



# スクーミーオレンジボード

単四電池1本で動き、センサーやアクチュエーターを選んで、  
つなげるだけで動く、小型コンピューターです。  
動作を指示するプログラムも、指令ブロックを組み合わせるだけで  
生成できるため、素早く簡単に思い通りの動作をさせることができるIoTツールが「スクーミー」です。

小学校・中学生といった初心者から、高等学校の科目となった「情報Ⅰ」「情報Ⅱ」「総合的な探究の時間」で応用的に  
活用されるなど、身の回りの課題から社会全体の課題解決まで、さまざまな場所で使われるツールになっています。



## 書き込み機 (書き込み機延長ケーブル)

スクーミーオレンジボードとパソコンを繋げて利用します。  
パソコンでプログラムを組み、そのプログラムをスクーミーオレンジボードに書き込む  
ためのツールです。

電源も供給するため USB から電源を取ることもできます。  
Unity や他のサービスと連携するとき (シリアル通信) も書き込み機を使うことで手軽  
に通信させる機能を持っています。  
書き込み機延長ケーブルを利用することでデータを取得するときやゲームのコントロー  
ラなどで利用することができます。



## センサーコネクタ

USB タイプで半田づけや電気回路の知識がなくてもつなげ  
るだけで利用することができます。  
様々な種類のセンサーコネクタがあり、1つのボードに  
同じものをつなげることもできますし、異なったものをつ  
なげることもできます



# はじめまして。 オレンジボードです。

School Myself。  
これが、私たち SchoMy (スクーミー) の語源です。

こうだったら良いのにな!とか、  
こんなこと出来ないのかな?と学校の休み時間に何気なく話している  
ことを、そこに行って、すぐに試してみることが出来るモノとして、  
このスクーミーボードは生まれました。

誰もが、身の回りにたくさんある小さな気づきや発見したことをその  
ままにしないで、電池や電球を取り換えるくらい簡単に、試して、改  
善しに行くことができるツールがこのスクーミーボードです。



## スクーミーブロックエディタ

ブロックを組み立てて直感的にプログラミングをすることができます。つくったプログラムを保存して開くこともできますし、ウェブページからダウンロードしたプログラム (sbp ファイル) をここで開くこともできます。シリアルモニタ・グラフモニタを利用することで、センサーコネクタの値をすぐにデータとして見るすることができます。

## スクーミー IDE

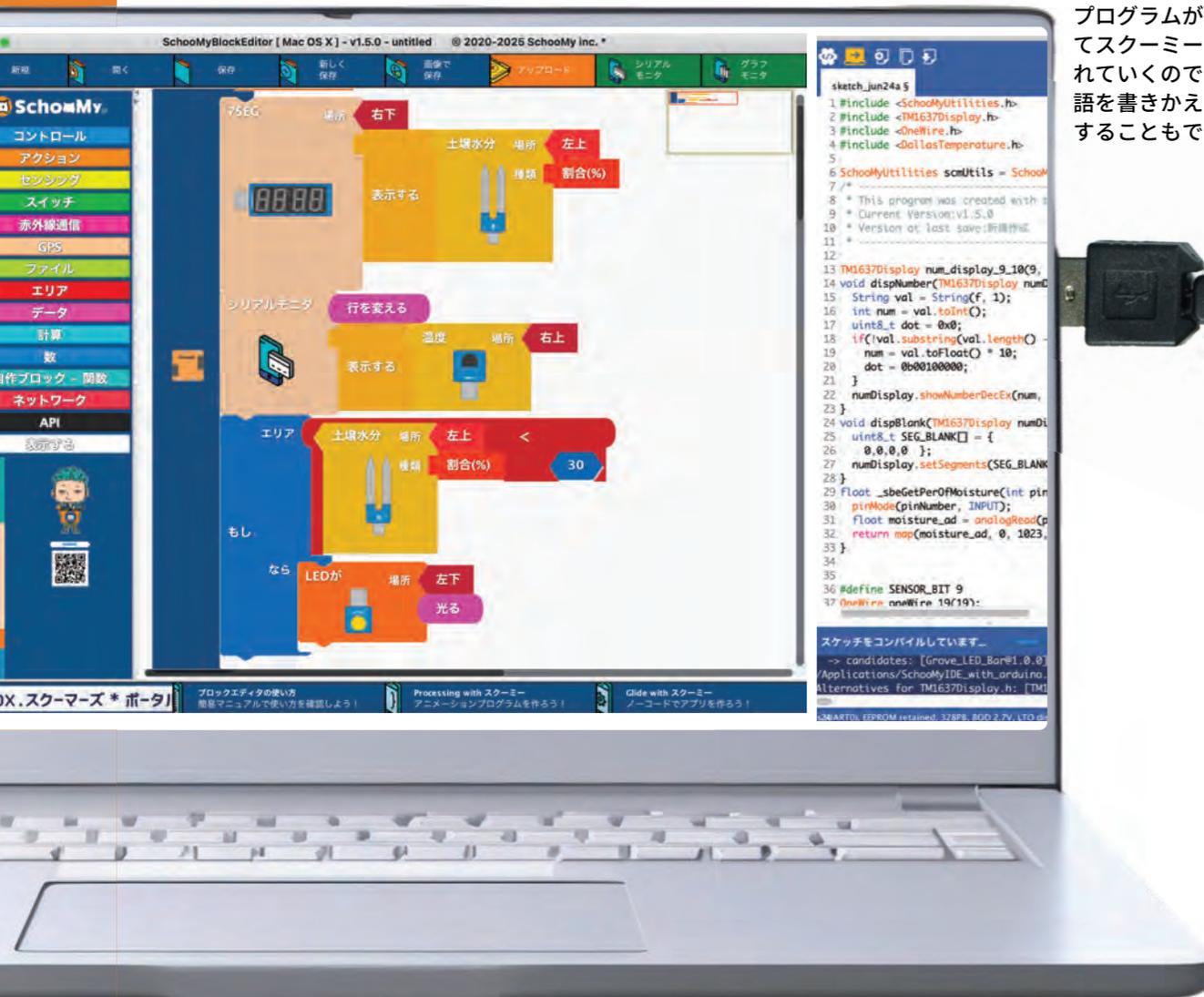
スクーミーボードに書き込むプログラムをつくる場所のことです。スクーミーブロックエディタでブロックを組み立てて、アップロードをおすことで、IDE 上で言語が生成されます。

プログラムが書き込み機を通じてスクーミーボードに書き込まれていくのです。もちろん、言語を書きかえてプログラミングすることもできます。

# 測って、比べる。 すぐに試すことができます。

- 本体の大きさは、縦 5 センチ、横 5 センチ
- センサーコネクタは利用者のやりたいことがすぐに試せるよう、温度センサーや明るさセンサーなど 50 種類以上が用意されている
- センサーおよびアクチュエーターを microUSB の端子接続で同時に最大 4 種類接続できる

- センサーやアクチュエーターは半田付けや電気回路の知識がなくても簡単に差し替えができる
- 給電は小型コンピュータに単四電池ソケットが付随しており、電池で動かす際に配線を必要としない
- 給電は本体内蔵の単四電池での給電以外に、コンピュータと接続することで microUSB 経由で給電、および AC アダプタからの給電も可能である
- スクーミー本体にネジ穴が空いていることで、制作物を板などと固定することができる
- スクーミーを用いて取得された温度データなどの情報は、スクーミー本体に取り付けた SD カードなどの外部記憶装置との接続により保存が可能
- 作成されたプログラムはスクーミー本体および学習端末に保存することができる。(なお、スクーミー本体に保存できるプログラムは 1 つ)



プログラムが書き込み機を通じてスクーミーボードに書き込まれていくのです。もちろん、言語を書きかえてプログラミングすることもできます。

## コネクタ延長ケーブル

センサーコネクタをのぼして使うことのできるコネクタ専用の延長ケーブルです。



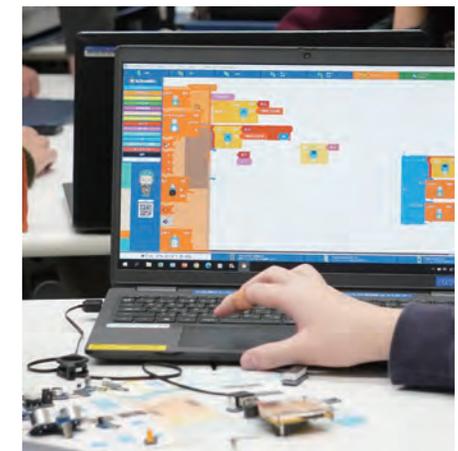
### 【動作環境】

- Windows、MacOS、Chromebook に開発環境のアプリケーションをインストールすることができ学習端末に対応できる
- 屋外でも利用できるように、開発環境がオフラインでも利用できる
- 開発言語は、独自のオリジナルのビジュアル言語でも可とするが、C 言語、Python、Javascript、に変換が可能である
- 開発環境であるブロックエディタは、日本語対応のほか、英語における学習にも対応できている
- 小型コンピュータの動作は、学習端末で作成したプログラムを Bluetooth 通信の接続で行うのではなく、小型コンピュータに内蔵されている IC チップに学習端末で生成したプログラムソースコードをアップロードし、動作させる

今、スクーミーボードは、教室から外に飛び出して行く子どもたちと一緒に、体育館や校庭、公園から畑、そして自宅やおじいちゃん家、病院や会社といった、あらゆる場所で多くの人に使われ始めています。

小学生が探究するためのコンピュータとして生まれたスクーミーボードが今や、半田付けもおこなずに、そのまま使える便利さが多くの方に受け入れられ、自ら、家庭や地域を改善するツールとして、世の中に新しく広まってきているのです。

スクーミーは、つくることで、すぐ解決できる世界をつくります。





# スクーミーブロックエディタのインストール

難易度 ★☆☆☆☆ 時間の目安 30分

## スクーミー IDE | 総合開発環境 とは

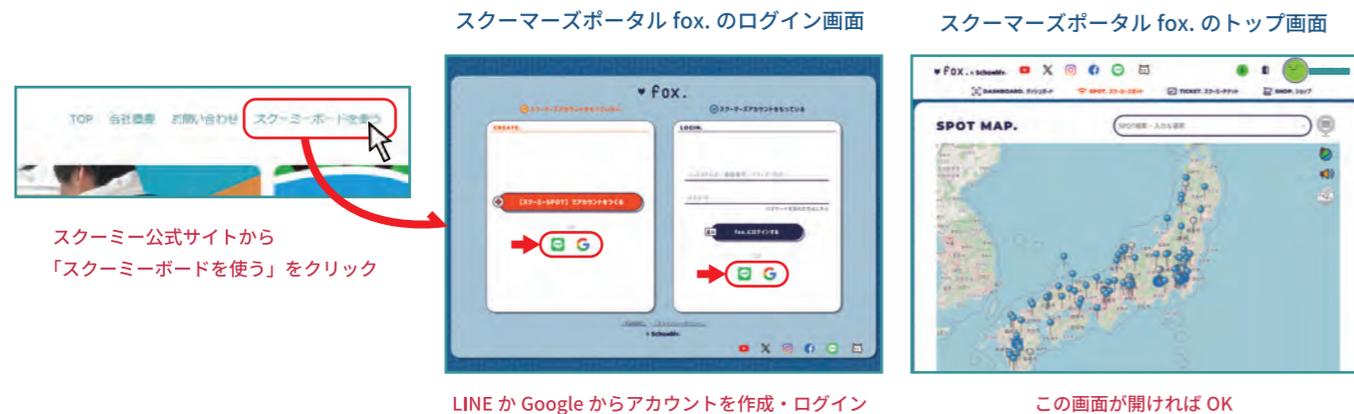
スクーミー IDE は、スクーミーボードにプログラミングをするためのソフトウェアです  
パソコンにスクーミー IDE とドライバをインストールして、スクーミーボードを使うための準備をしましょう

01 インターネットで「スクーミー」と検索して、スクーミーの公式サイトにアクセスします



02 スクーミーの公式サイト右上にある「スクーミーボードを使う」をクリックしてスクーマーズポータル fox. を開きます

03 スクーマーズポータル fox. を開いたら、「スクーマーズアカウントをもっていない」方は、LINE か Google からアカウントを作成してください  
既に「スクーマーズアカウントをもっている」方はログインして、スクーマーズポータル fox. のトップ画面を開きましょう



- 04 スクーマーズポータル fox. のページ上部にある「SPOT. スクーミースポット」にカーソルを合わせると、メニューが出てきます  
「▶ MANUAL. マニュアル」をクリックして、マニュアルページを開きます
- 05 「開発するための環境をつくる」をクリックして、スクーミー IDE の説明を読みましょう  
自分のパソコンの OS に合った版のインストールページに移動します ※自分のパソコンの OS を調べておきましょう



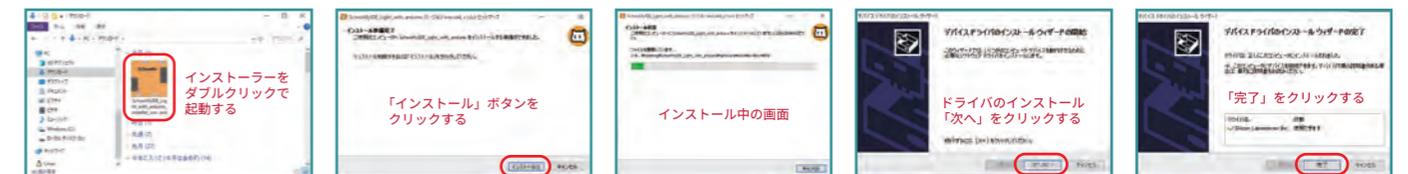
06 インストールページに移動したら、手順を読みすめ下にスクロールします  
インストーラーのダウンロードボタンをクリックしましょう

※インストーラーには「通常版」と「軽量版」があります  
授業や部活動で使用する場合は、軽量版をおすすめします

07 インストーラーのダウンロードができたなら  
インストーラーを起動します (画像は Windows のものです)

※セキュリティソフトからの警告がでる場合がありますが  
問題ありませんので、実行してください

インストーラーの指示に従って、  
スクーミー IDE とドライバーのインストールしましょう



07 左の画面が表示されたらインストールができました  
「完了」ボタンをクリックして、スクーミー IDE が起動するのを確認しましょう

Windows の場合は、デスクトップ画面に  
右の 2 つのアイコンができていれば完了です

これで開発環境を準備することができました!



LEVEL  
01

スクーミーボードを使うための準備

スクーミーを使うための準備をする

スクーミーボードについて知ろう

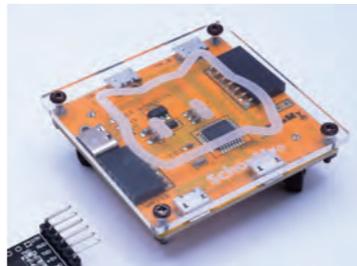
難易度 ★☆☆ 時間の目安 05分

目標 スクーミーボードの方向を覚える

SchooMy Start Guide Book は、初めてスクーミーボードを使う人向けのガイドブックです  
スクーミーボードやブロックエディタについて知り、スクーミーボードを使うための準備をしましょう

01 スクーミーボードを手にとってみましょう

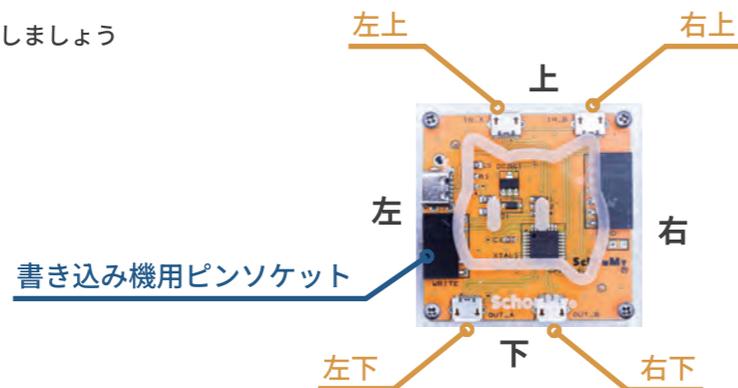
スクーミーボードは、5cm 四方の小型コンピュータ スクーミーボードであり、  
プログラムを書き込み、センサーを取り付けることで使うことができます



02 スクーミーボードには方向があります

右の画像を見ながら、スクーミーボードの方向を確認しましょう

- スクーミーボードの表と裏、上下左右の方向
- 左上、右上、左下、右下の micro USB の場所
- 書き込み機用ピンソケットの場所



ここまでできたら LEVEL01 クリア!

スクーミーボードに触れて、  
スクーミーボードの方向を理解できた

LEVEL  
02

SchooMyIDE とブロックエディタの立ち上げ

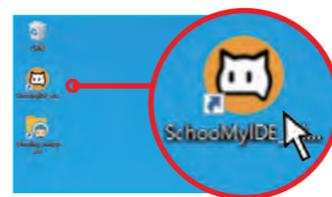
スクーミーを使うためにパソコンに環境をつくる

パソコンでスクーミーブロックエディタを開こう

難易度 ★☆☆ 時間の目安 05分

目標 スクーミーブロックエディタを開いて、画面に見やすく配置する

01 SchooMyIDE\_with\_arduino のアイコンをダブルクリックして SchooMyIDE を起動します

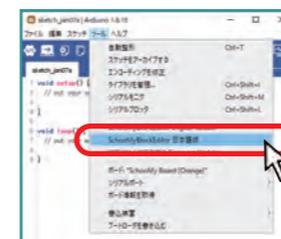


SchooMyIDE 起動中の画面

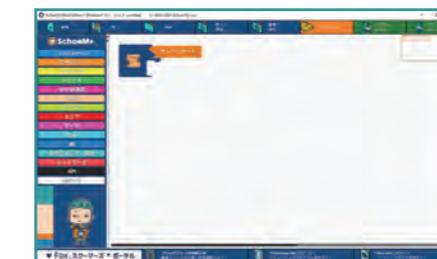


SchooMyIDE の画面

02 SchooMyIDE の [ツール] メニューから [SchooMyBlockEditor 日本語版] をクリックしてブロックエディタを起動します



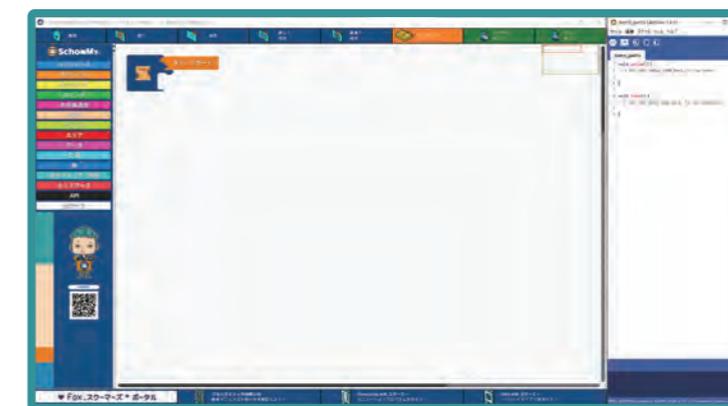
SchooMyIDE の画面



ブロックエディタの画面

03 ウィンドウの位置と大きさを調整して、  
右画像のような配置にします

SchooMyIDE は横幅を狭くして右側に、  
ブロックエディタは大きくして左側に配置します



ここまでできたら LEVEL02 クリア!

スクーミーブロックエディタを開いて、  
画面に見やすく配置できた

LEVEL  
03

スクーミーボードとパソコンの接続

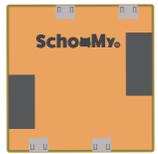
スクーミーボードとパソコンを接続する

スクーミーボードとパソコンを繋げて動作を確認しよう

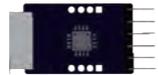
難易度 ★☆☆ 時間の目安 10分

目標 スクーミーボードとパソコンを接続し、接続されたことを確認する

準備するもの



スクーミーボード

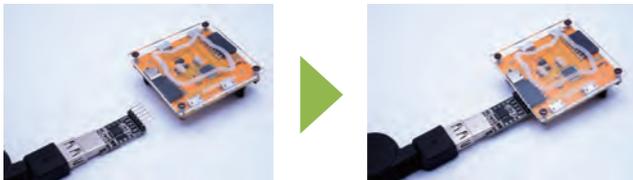


書き込み機



パソコン または タブレット

01 スクーミーボードに書き込み機を挿しこみます



書き込み機の挿す場所、挿す方向に気をつけましょう

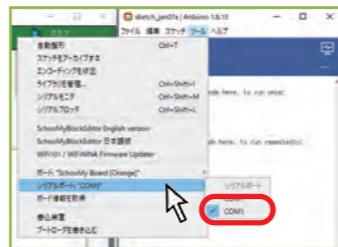
02 パソコンに書き込み機を挿しこみます



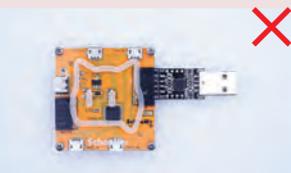
スクーミーボードとパソコンを接続すると、書き込み機が光ります

03 SchooMyIDE の [ ツール ] メニューから [ シリアルポート ] にカーソルを合わせます

[ COM0 ] や [ COM1 ] などが表示され、✓ が付いていることを確認します



正しく表示されない場合や、書き込みがうまくいかない時は  
スクーミーボードと書き込み機がしっかりつながっているかを確認しましょう



書き込み機を挿す場所を間違えている



書き込み機を挿す向きを間違えている



書き込み機のピンの位置がずれている

ここまでできたら  
LEVEL03 クリア!

スクーミーボードとパソコンを接続し、  
シリアルポートが表示された

LEVEL  
04

スクーミーボードでシリアルモニタを使った表示

シリアルモニタに文字を表示させる

シリアルモニタの使い方を知らう

難易度 ★☆☆ 時間の目安 10分

目標 シリアルモニタを開いて、文字を表示することができる

01 ブロックエディタを開き、[ 表示する ] タブの  
シリアルモニタブロックを出してつなげます

ブロックの凹凸が合うようにつなげましょう  
うまくブロックがつながるとカチッと音がなります



02 ブロックエディタの [ アップロード ] をクリックして、  
ボードへ書き込みます



ボードへの書き込みが完了しました。

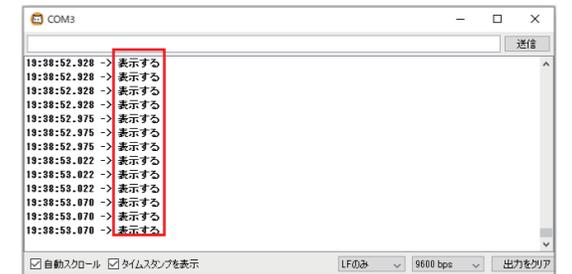


アップロード時に下のようなエラーが出る場合があります  
「スクーミーボードが動かない場合の対応について」のページをご覧ください  
スケッチの書き込み中にエラーが発生しました エラーメッセージをコピーする

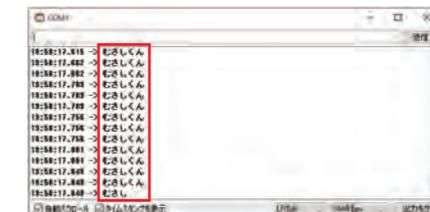
03 ブロックエディタの [ シリアルモニタ ] をクリックして、  
シリアルモニタを出します



「表示する」と文字列が表示されているのを確認しましょう  
確認したら、シリアルモニタの [ X ] ボタンをクリックして閉じます



04 シリアルモニタブロックの黄色い部分をクリックして、文字を打ちこんでみましょう  
[ アップロード ] して、[ シリアルモニタ ] に表示される文字が変わったのを確認しましょう



ここまでできたら  
LEVEL04 クリア!

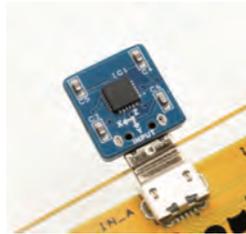
シリアルモニタに文字を表示できた  
表示する文字を変えることができた

**LEVEL 05** スクーミーボードで角度センサーを使ったセンシング  
**角度センサーの反応を見る**  
 角度センサーを使ってみよう

難易度 ★★★ 時間の目安 10分

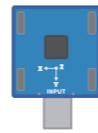
**目標** 角度センサーを傾けて、シリアルモニタの値が変化することを確認する

01 スクーミーボードの左上に角度センサーを挿します



準備するもの

角度センサー



02 [センシング]のタブから角度センサーブロックを出して、シリアルモニタブロックの右側につなげます



ブロックの凹凸が合うように繋げましょう



上のようブロックが繋がれば完成です

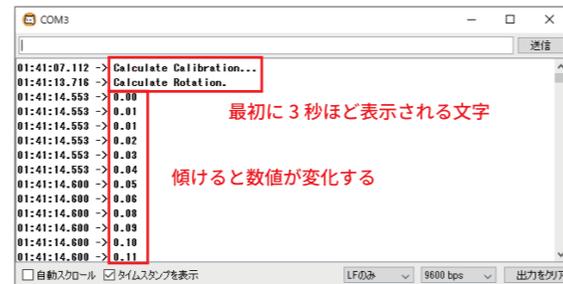
04 ブロックエディタの[アップロード]をクリックして、ボードへ書き込みます  
 ブロックエディタの[シリアルモニタ]をクリックして、シリアルモニタを出します



05 角度センサーがついたスクーミーボードをいろいろな方向に傾けて、シリアルモニタの数値が変わることを確認しましょう

ここまでできたら LEVEL05 クリア!

角度センサーをいろいろな方向に傾けて、シリアルモニタの数値が変化することを確認できた



NEXT ▶▶▶ 角度センサーを傾けたら文字を変化させる

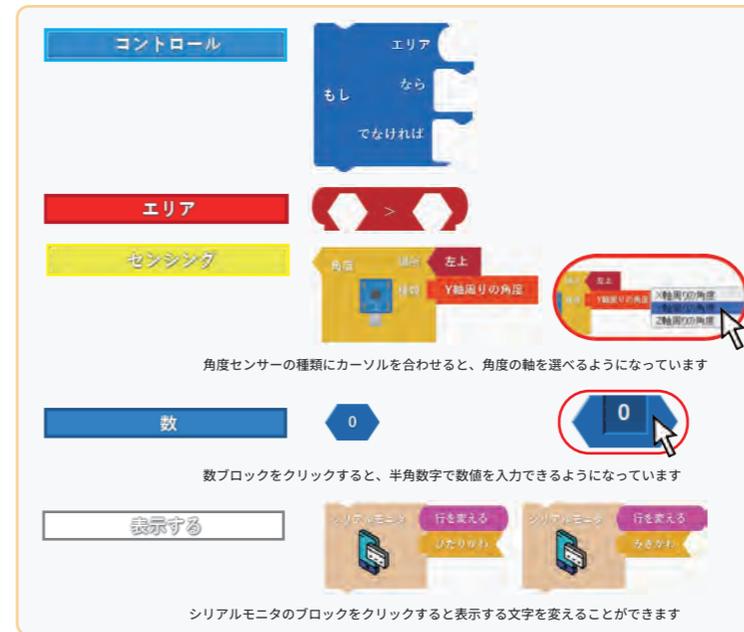
**LEVEL 06** 角度センサーと条件分岐を使ったシリアルモニタのコントロール  
**角度センサーを傾けたら文字を変化させる**  
 もし~なら~ブロックの使い方を知ろう

難易度 ★★★ 時間の目安 15分

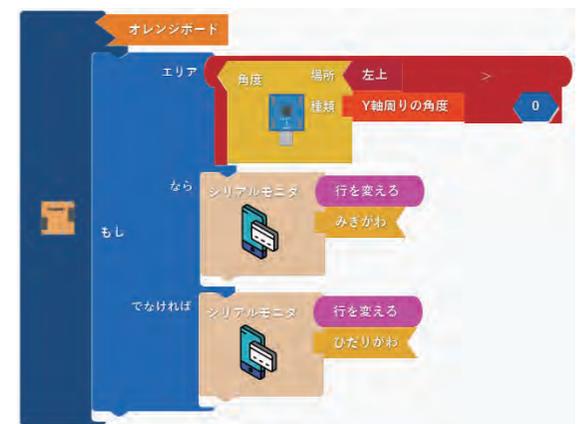
**目標** 角度センサーを左右に傾けたら、シリアルモニタの文字が変化する

01 スクーミーボードの左上に角度センサーが挿されていることを確認しましょう

02 それぞれのタブのメニューから、以下のブロックを出します



03 下の画像のようにブロックを並べます



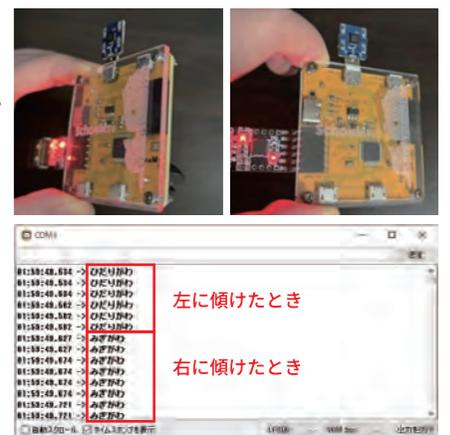
04 [アップロード]をクリックしてボードへ書き込み、[シリアルモニタ]をクリックして、シリアルモニタを出します

05 「Calculate Calibration...」が表示されている間は、動かさずに水平に保ちます  
 「Calculate Rotation」と表示されたら、スクーミーボードを左右に傾けてみましょう

スクーミーボードを左に傾けた時は「ひだりがわ」と表示され、スクーミーボードを右に傾けた時は「みぎがわ」と表示されるのを確認しましょう

ここまでできたら LEVEL06 クリア!

角度センサーを左右に傾けて、シリアルモニタの表示が変化するのを確認できた



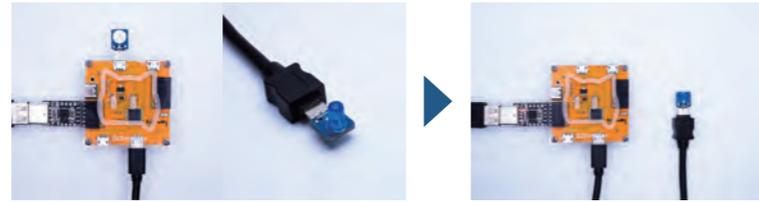
NEXT ▶▶▶ スクーミーボードの活用方法

## スクーミーボードの活用方法

### 01 コネクタ延長ケーブルを使う方法

コネクタ延長ケーブルを使うことで、スクーミーボードから離れたところでセンサーを使うことができます  
ものづくりをするときに使ってみましょう

※コネクタ延長ケーブルは別売りです



### 02 スクーミーボードを単四電池で動かす方法

スクーミーボードの裏面にある電池ボックスに単四電池を入れることで、パソコンに接続せずにスクーミーボードを動かすことができます

- スクーミーボードに書き込まれたプログラムが動いていることを確認します
- パソコンから書き込み機を抜き、スクーミーボードからも書き込み機を抜きます
- スクーミーボードを裏返し、プラスマイナスの向きに従って単4電池を取り付けます
- スクーミーボードの裏面のスイッチを [BOX] にするとスクーミーボードが電池で起動します

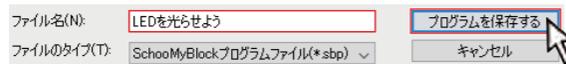


※プログラムをなおすためにパソコンに再接続するときは、必ずスイッチが [PRG USB] に入っていることを確認してください

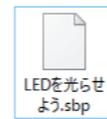
### 03 作ったプログラムをパソコンに保存する方法



[新しく保存] をクリック  
※上書き保存の場合は [保存] をクリック



ファイル名を入力して [プログラムを保存する] をクリック

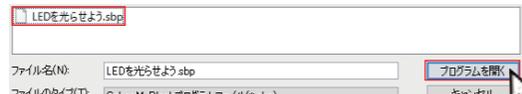


.sbp ファイルが完成

### 04 保存したプログラムを開く方法



[開く] をクリック



.sbp ファイルを選択して [プログラムを開く] をクリック

### 05 新しいプログラムを作成する方法

新しいプログラムを作成するときは、ブロックエディタの [新規] をクリックします



### 06 ブロックエディタのブロックを複製・削除する方法

ブロックエディタのブロックの上で右クリックをすると、ブロックの複製や削除ができます  
ブロックをドラックし画面外でドロップすることでも、ブロックを削除できます



## スクーミーボードが動かない場合の対応について

### 01 ブロックエディタの [アップロード] をクリックした時に「エラーが発生しました」とメッセージが表示され、正常に書き込めない

- スクーミーボードと書き込み機がしっかりつながっていて、パソコンにもつながっているのかを確認する



書き込み機を挿す場所を間違えている



書き込み機を挿す向きを間違えている



書き込み機のピンの位置がずれている



裏面のスイッチが PRGUSB ではなく BOX に入っている  
※PRGUSB の方に入れてください

- 正しくつながっているのにエラーが出る場合は、スクーミーボードをリセットする

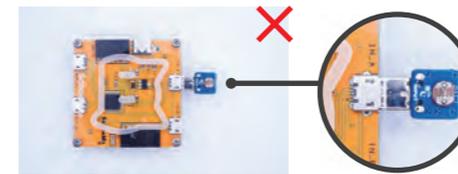


スクーミーボードの裏面のリセットボタンを指で押しながアップロードをクリックして、「マイコンボードに書き込んでいます...」の表示になったら指を離すことでリセットできます

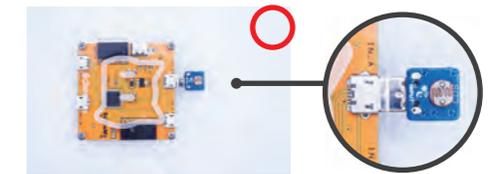
- SchooMyIDE を再起動する

### 02 エラーは発生せず書き込みはできるが、プログラム通りに正常に動作しない

- スクーミーボードに接続しているコネクタが壊れていないか、正しく接続されているか確認する



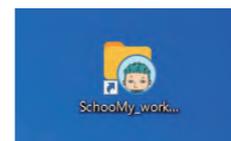
センサーが奥まで挿さっていない



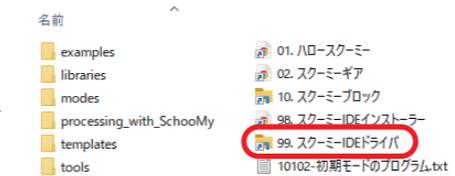
センサーが奥までしっかり挿しこんである

### 03 アップロードができない、スクーミーボードがパソコンに認識されない

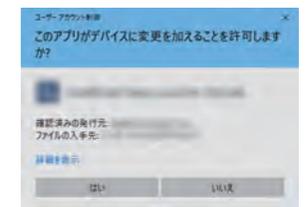
- ドライバーがパソコンにインストールされているか確認する ※Windows のみの対処法



SchooMy\_workspace フォルダを開く



99. スクーミー IDE ドライバをダブルクリックする

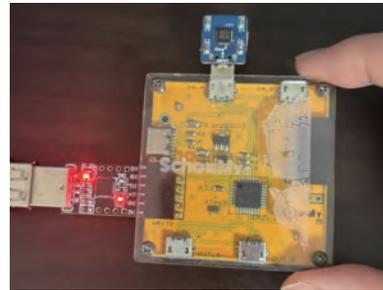


「はい」を選び、手順に沿ってドライバーをインストールする

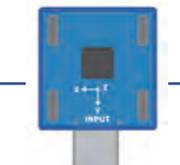
# スクーミーボードと加速度（角度）センサーコネクタの準備をする

スクーミーボードで取得した角度の値を元にゲームのキャラクターが動くようにする準備をします。

## ▼ スクーミーボード

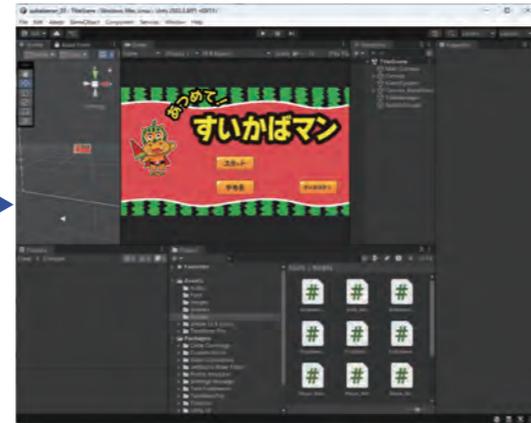


使用するセンサ



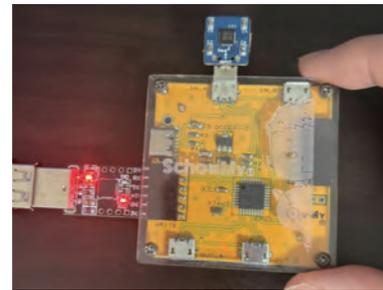
加速度（角度）センサーコネクタ

## ▼ Unity の画面



加速度（角度）センサーで取得した「y」の値が表示されることを確認しましょう

## ▼ スクーミーボード



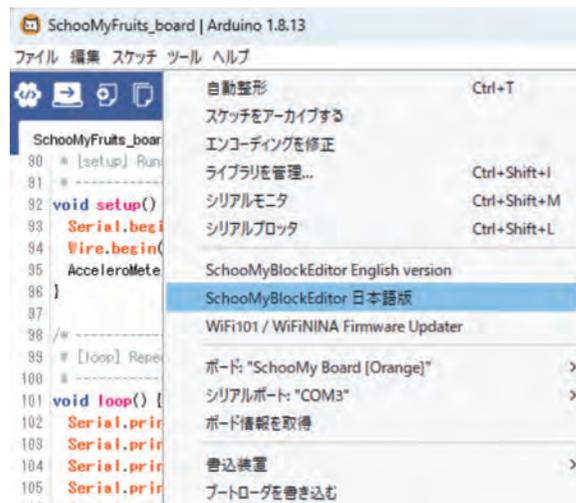
## ▼ シリアルモニタ



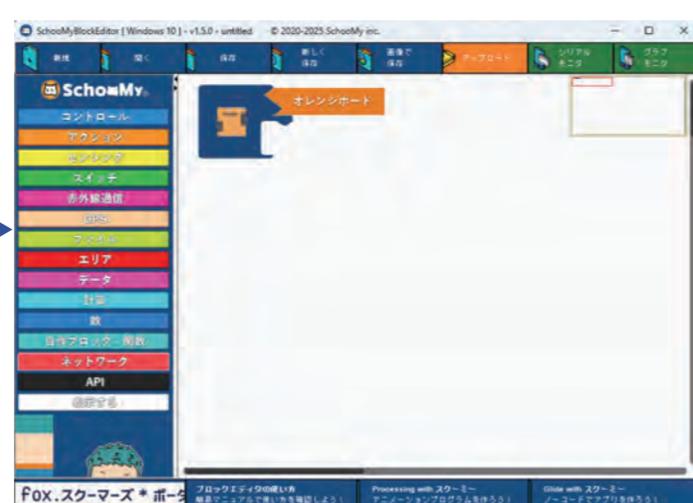
## 設定方法

①スクーミー IDE から、スクーミーブロックエディタを起動させます

### ▼ スクーミー IDE

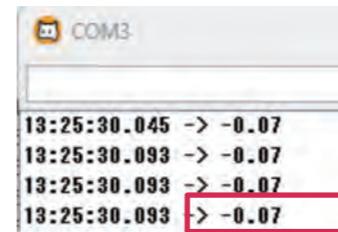


### ▼ スクーミーブロックエディタ



②シリアルモニタに「y」の値が表示されるブロックコードを作成します

### ▼シリアルモニタ

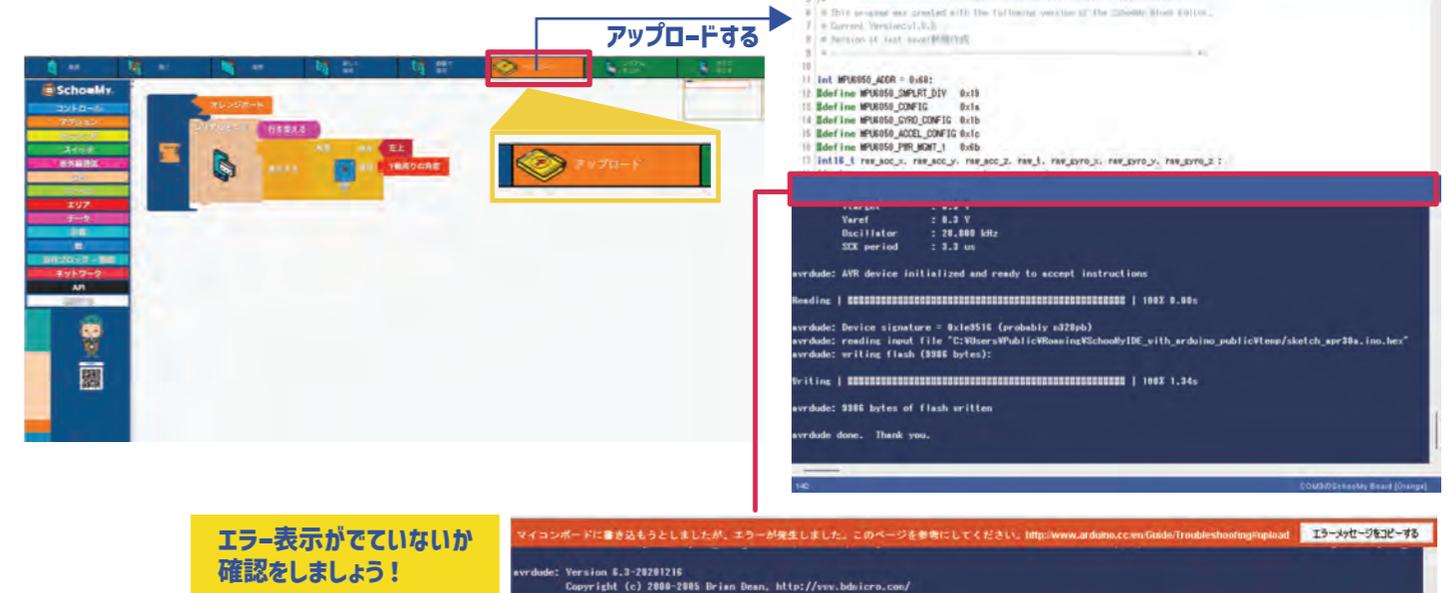


### ▼ブロックコード



スクーミーボードにプログラムを書き込んで結果を確認しましょう。

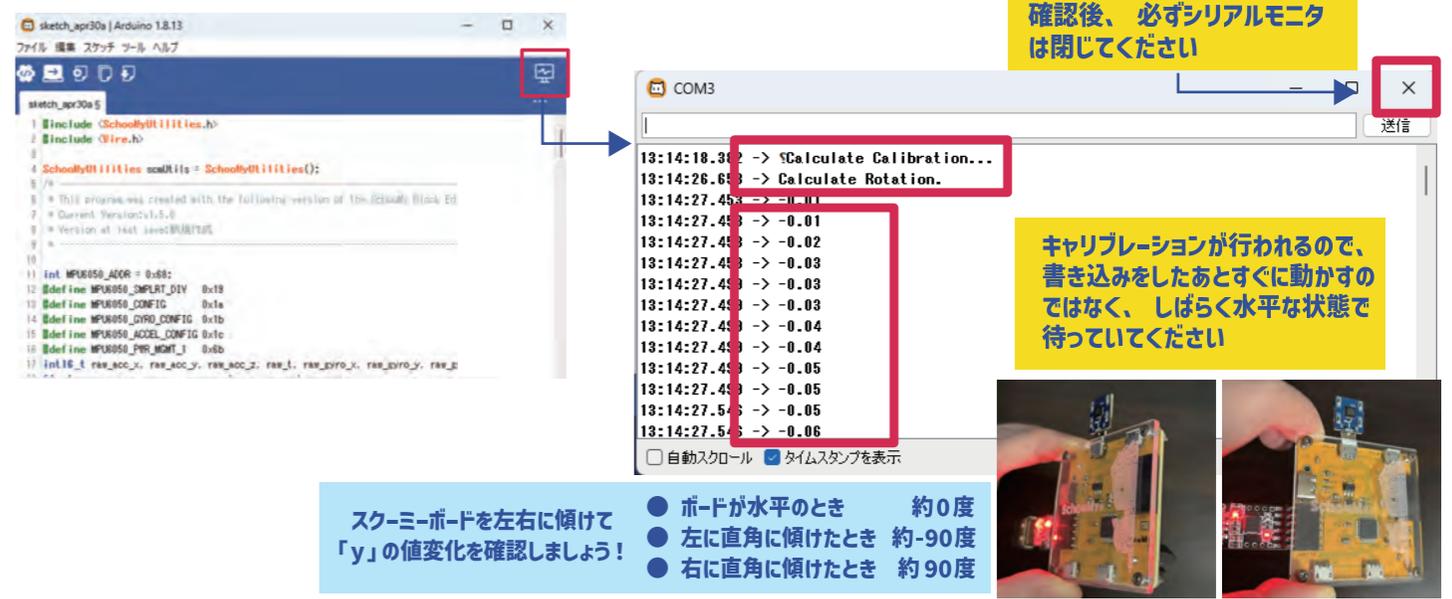
③「アップロード」をクリックし、スクーミーボードにプログラムを書き込みます



エラー表示がでていないか確認をしましょう！

マイコンボードに書き込もうとしたのですが、エラーが発生しました。このページを参照してください。 <http://www.arduino.cc/en/Guide/Troubleshooting/usbload> エラーメッセージをコピーする

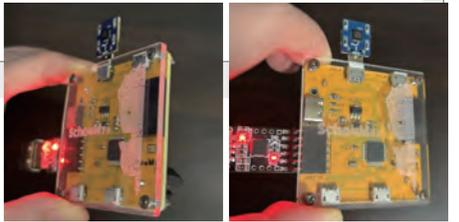
④シリアルモニタを表示させ「y」の値が表示されることを確認し、シリアルモニタを閉じます



確認後、必ずシリアルモニタは閉じてください

キャリブレーションが行われるので、書き込みをしたあとすぐに動かすのではなく、しばらく水平な状態で待っててください

- スクーミーボードを左右に傾けて「y」の値変化を確認しましょう！
- ボードが水平のとき 約0度
- 左に直角に傾けたとき 約-90度
- 右に直角に傾けたとき 約90度



# Unity の環境設定をする①

Unityで『あつめて!すいかばマン』を動かしましょう!

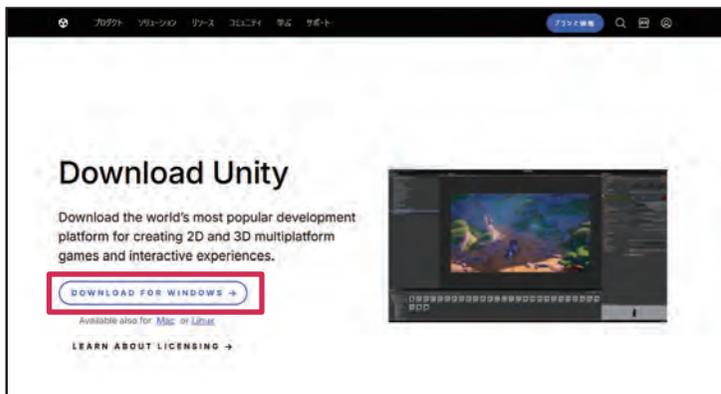
Unity の画面 ▶

スタートを押すと  
ゲームが開始されるようになります!

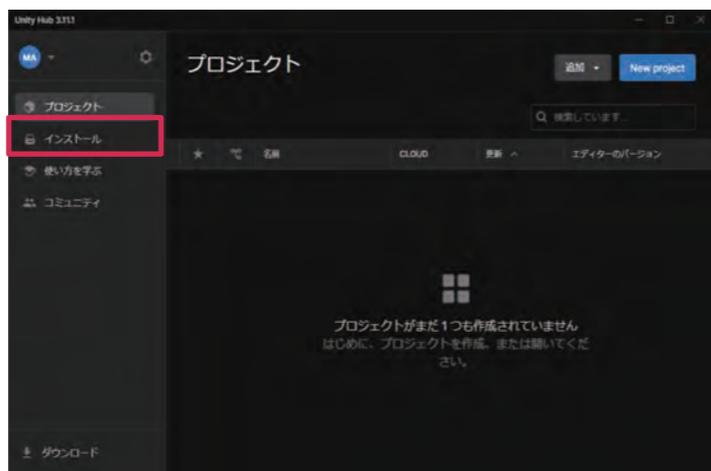


## Unityの準備 (Windowsの場合)

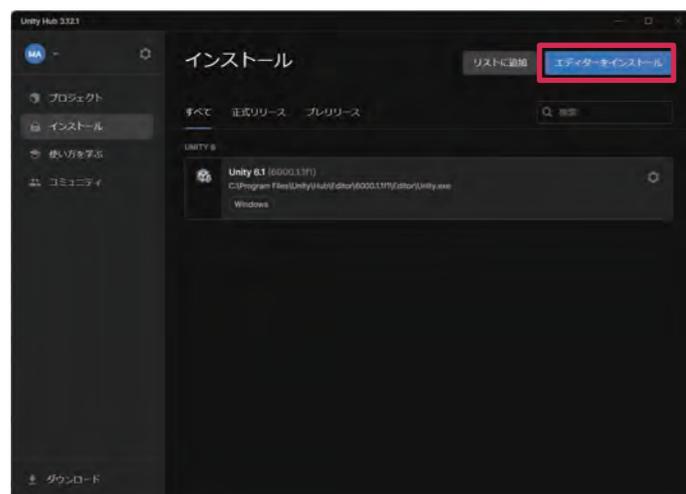
①公式サイトにアクセスし UnityHub をインストールします  
URL:<https://unity.com/ja/download>



②UnityHub を起動させ、「インストール」をクリックします



③「エディターをインストール」をクリックします



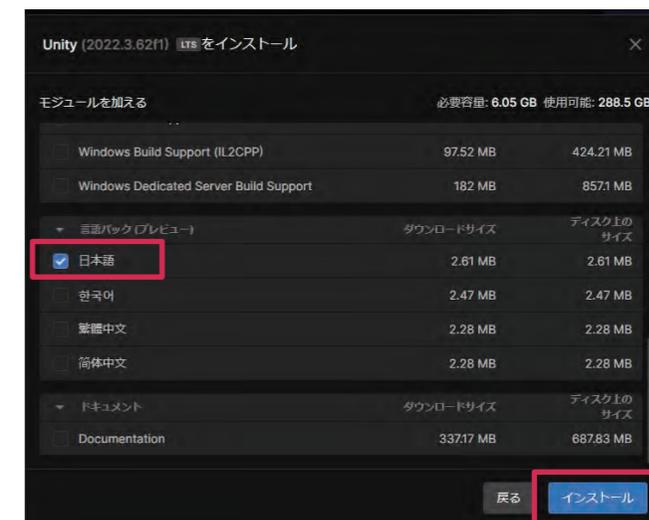
④「その他のバージョン」の Unity (Unity 6.x でない方) の「インストール」ボタンをクリックする  
※数字が大きく、 LTS がついていつものを選ぶことを推奨



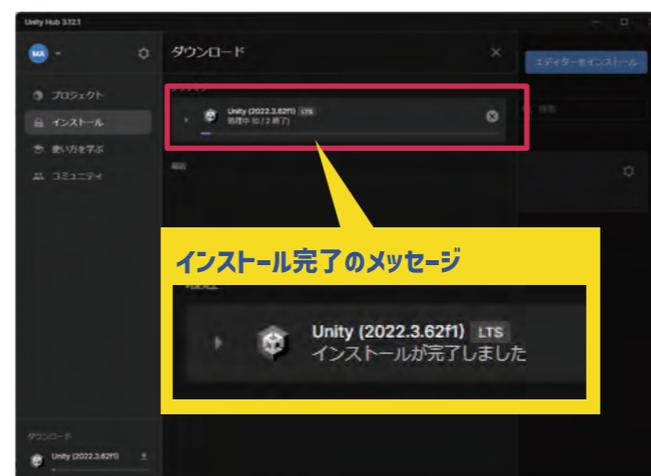
⑤モジュール追加画面が開くので、一番下までスクロールします



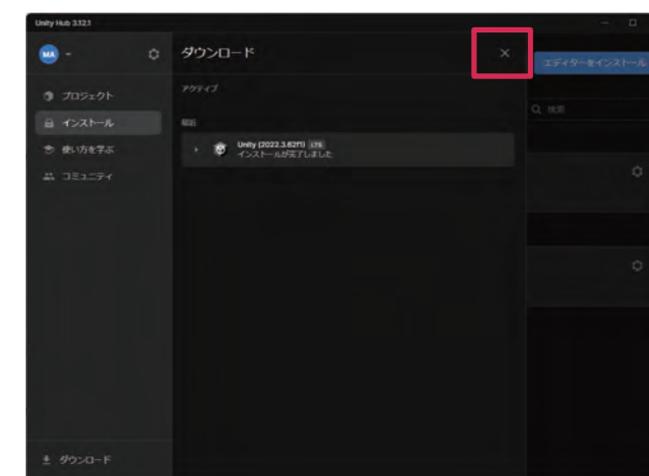
⑥「日本語」のみのチェックを入れ、「インストール」をクリックする



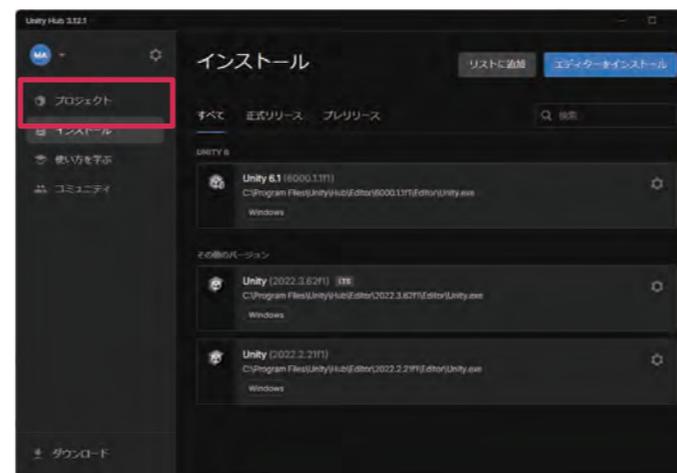
⑦ダウンロード & インストールが始まるのでしばらく待ちます  
※環境にもよりますが5~20分程度で完了します



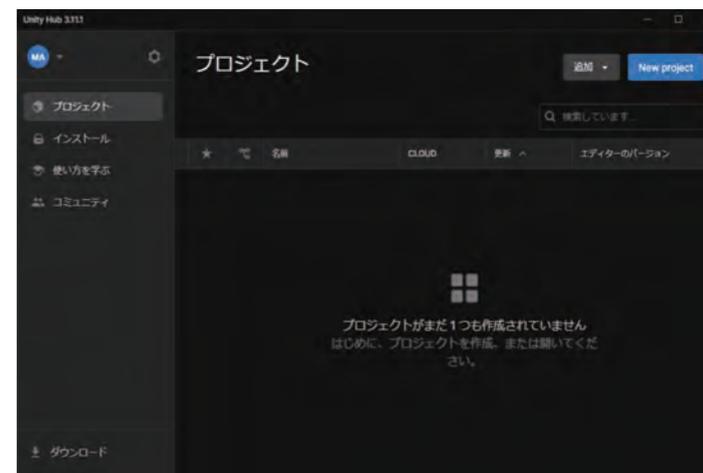
⑧インストールが完了したら「×」をクリックします



⑨「プロジェクト」をクリックします



⑩プロジェクトの画面が表示されます



## Unity の環境設定をする②

①プロジェクト画面で「New project」をクリックします



②「コア」を選択して、検索で「3D」を入力します  
その後、「3D (Built-In Render Pipeline)」を選択し  
「テンプレートをダウンロード」をクリックしてください



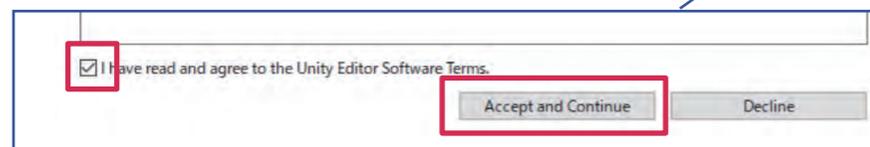
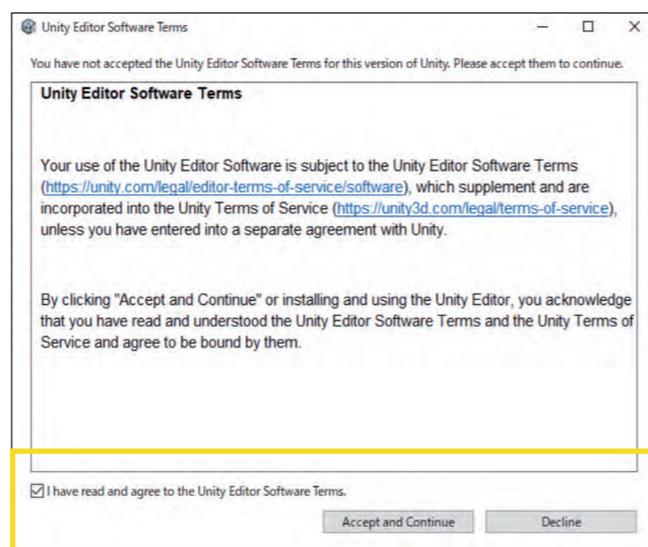
この画面が表示されないときは、次に進んでください  
(初回の設定時のみに表示されます)

③「3D (Built-In Render Pipeline)」を選択して、  
「プロジェクトを作成」をクリックします  
プロジェクト名や保存場所は任意で変更してください



最新のバージョンではなく、  
2022.x.xxxx  
を利用してください

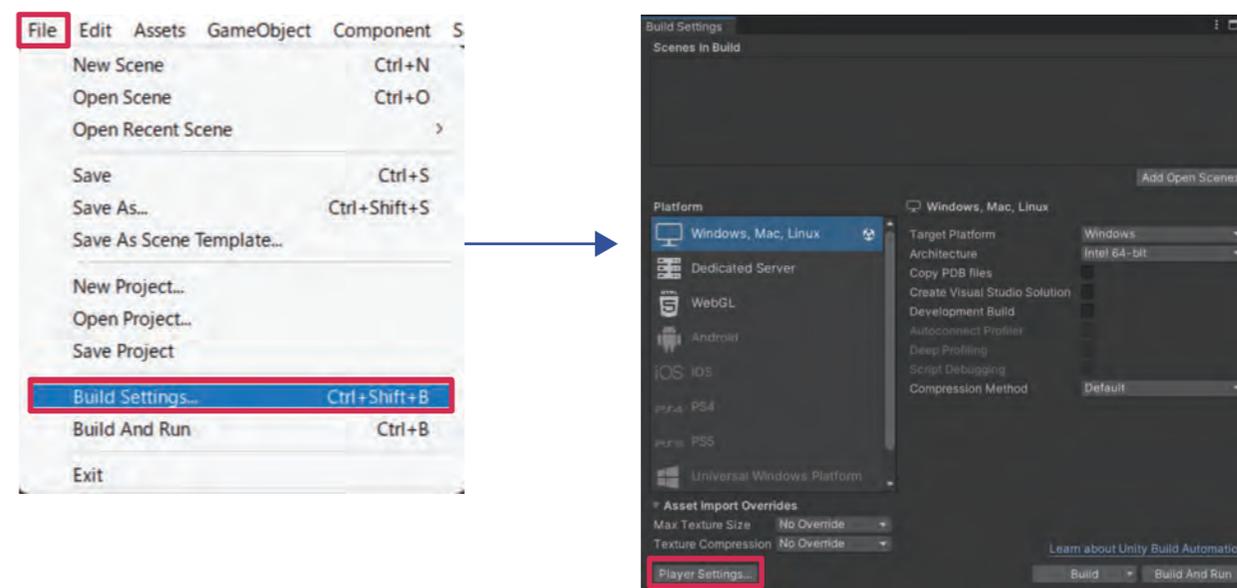
④Unity エディターのソフトウェア利用規約への同意を行います  
黄色枠の「」をチェックして、「Accept and Continue」  
をクリックしてください



この画面が表示されないときは、次に進んでください  
(初回の設定時のみに表示されます)

ビルド設定で .NET Framework を選択する

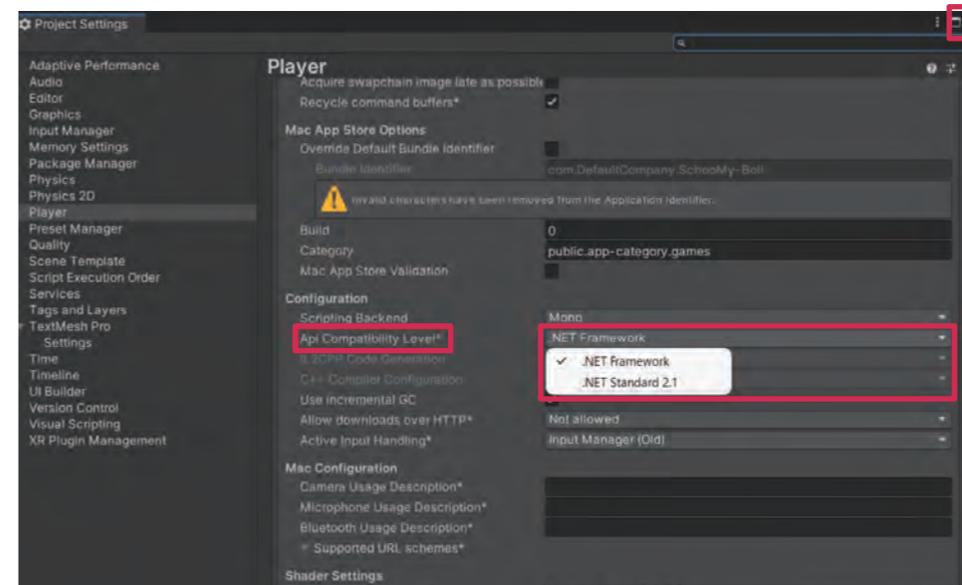
①File → Build Settings → Player Settings をクリックします



②API 互換性レベルを変更します

Other Settings」の真ん中あたりにある「Api Compatibility Level\*」を  
「.NET Standard 2.1」から「.NET Framework」に変更します

③変更後、  
画面を閉じます

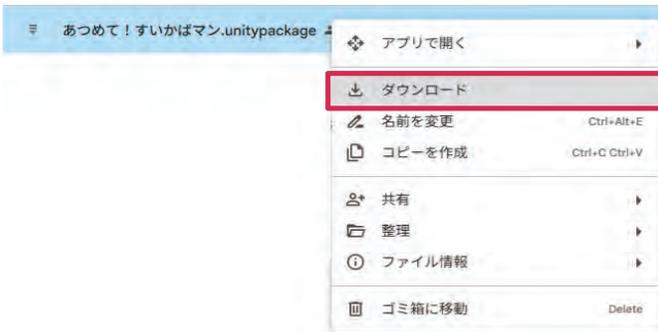


# 「あつめて! すいかばマン」をダウンロードして環境設定をする

あつめて! すいかばマンのUnityパッケージファイルをインポートします

①googleドライブにアクセスし「あつめて! すいかばマン .unitypackage」をダウンロードします

ホームページに掲載されている「DIVE- みんなのダイブ -」から、「特別号 Unity でゲーム開発をしよう」を開くと、noteの中にGoogleドライブへのリンクがあります。



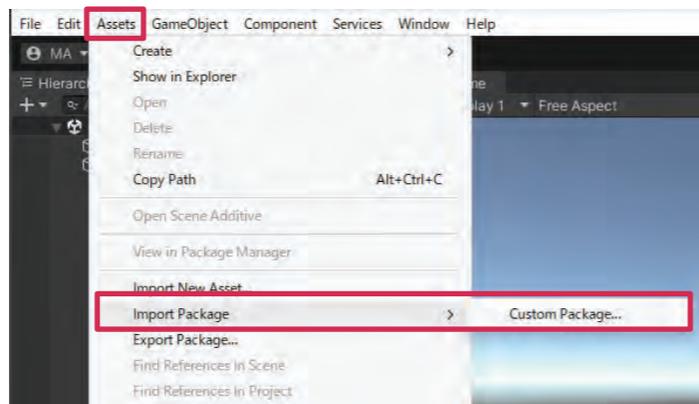
③あつめて! すいかばマン .unitypackage を選択して開きます



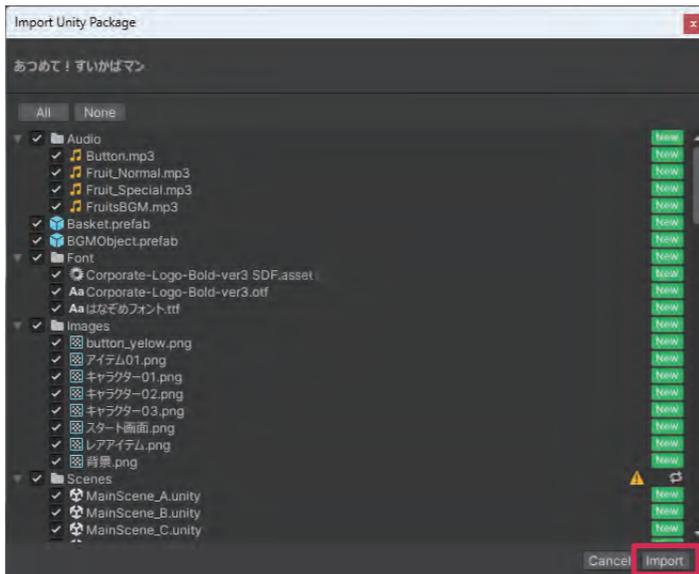
⑤「Assets」にファイルが追加されていることを確認します。



②Assets → Import Package → Custom Package でインポートファイルを選択する画面を開きます

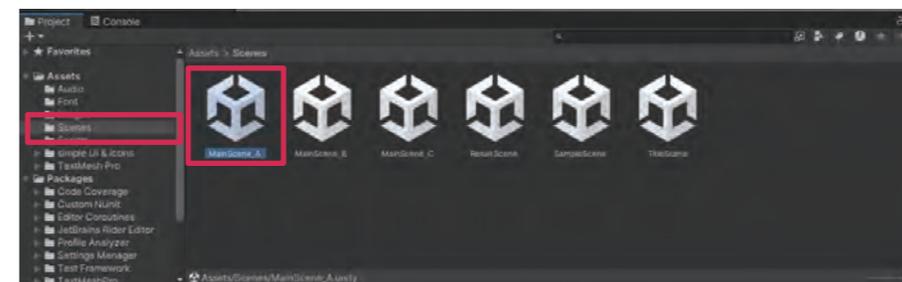


④「import」をクリックして Assets に追加します

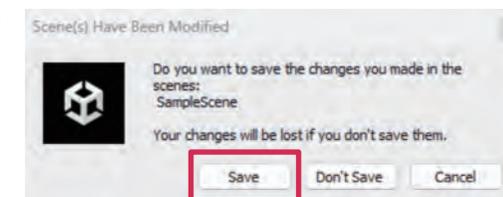


あつめて! すいかばマンのゲーム画面表示を設定します

①「Scenes」を読み込みましょう。「Scenes」を選択し「MainScene\_A」をダブルクリックします

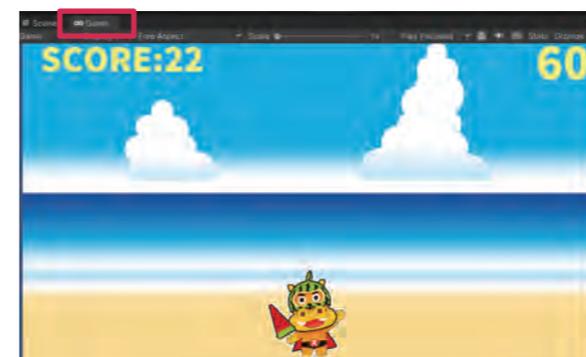


②保存(セーブ)を促す画面が表示されるので「Save」をクリックします

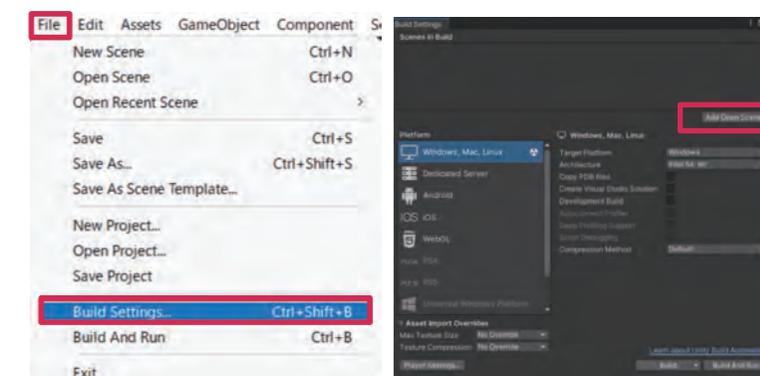


※インポート後、初回のみ表示されます

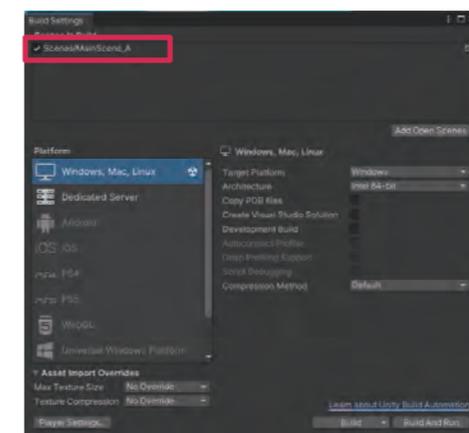
③Game タブに変更すると選択したScenesが表示されます



④File → Build Settings→Add Open Scenes をクリックして選択したスクリーンを追加します



⑤「Add Open Scenes」をクリックして、選択したスクリーンが写っていることを確認します。

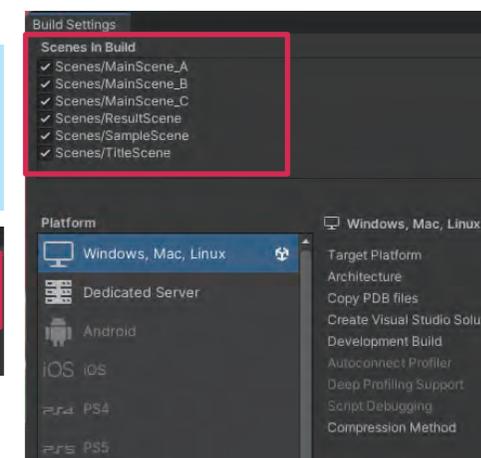


⑥Buildへ残り5つのScenesも同様に追加します



ダブルクリックで対象の Scenes を開いた後、Player Settings で「Add Open Scenes」の追加を5つ分、繰り返します。

⑦6つの「Scenes」が追加されていることを確認します(確認後、設定画面は閉じる)

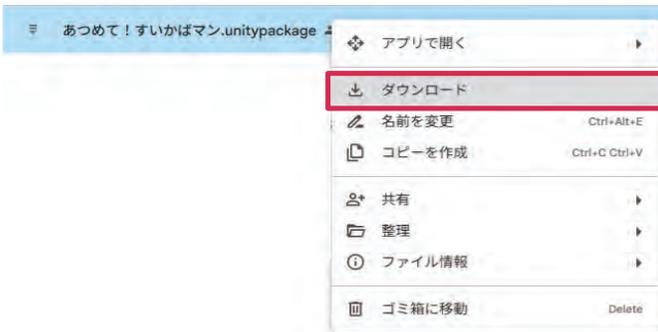


# 「あつめて! すいかばマン」をダウンロードして環境設定をする

あつめて! すいかばマンのUnityパッケージファイルをインポートします

①googleドライブにアクセスし「あつめて! すいかばマン .unitypackage」をダウンロードします

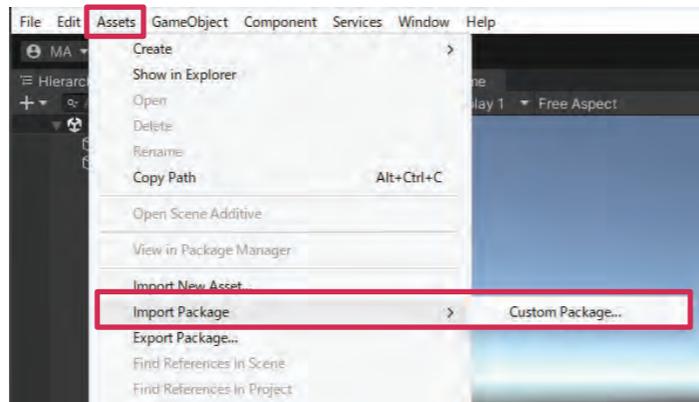
ホームページに掲載されている「DIVE- みんなのダイブ -」から、「特別号みんなのダイブ Unity でゲームをつくらう」を開くと、Googleドライブへのリンクがあります。



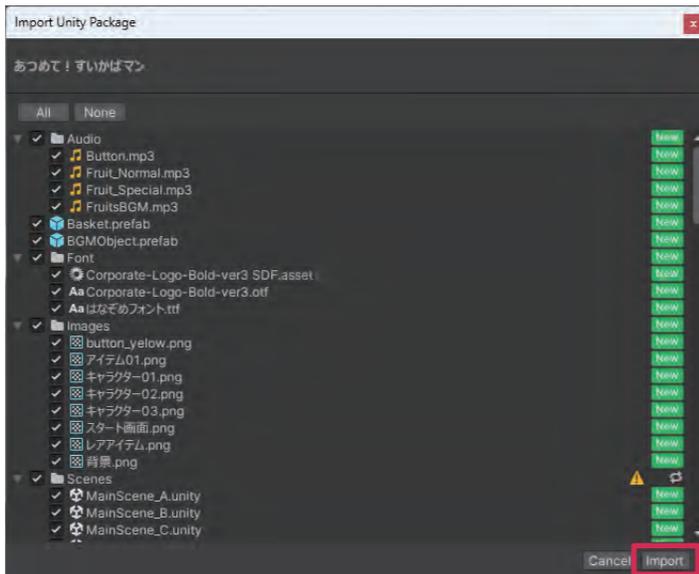
③あつめて! すいかばマン .unitypackage を選択して開きます



②Assets → Import Package → Custom Package でインポートファイルを選択する画面を開きます



④「import」をクリックして Assets に追加します

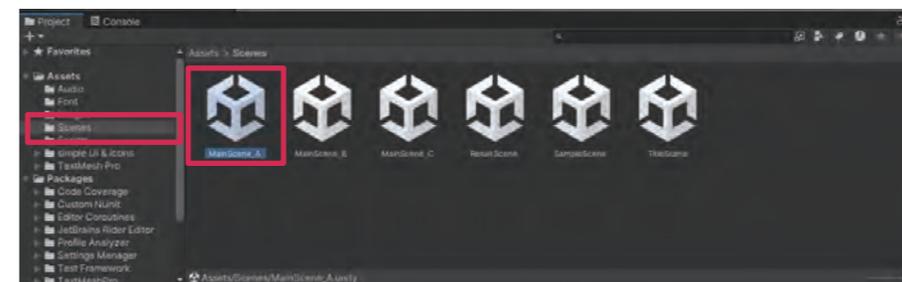


⑤「Assets」にファイルが追加されていることを確認します。

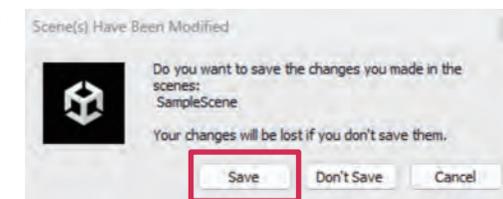


あつめて! すいかばマンのゲーム画面表示を設定します

①「Scenes」を読み込みましょう。「Scenes」を選択し「MainScene\_A」をダブルクリックします

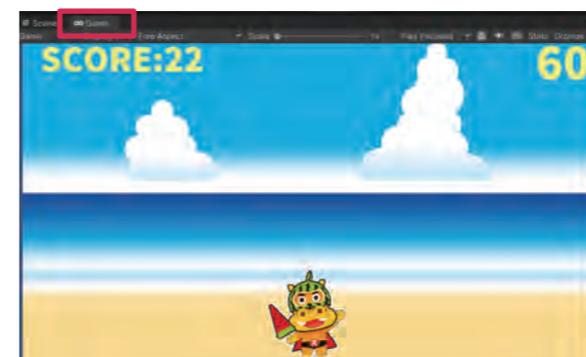


②保存(セーブ)を促す画面が表示されるので「Save」をクリックします

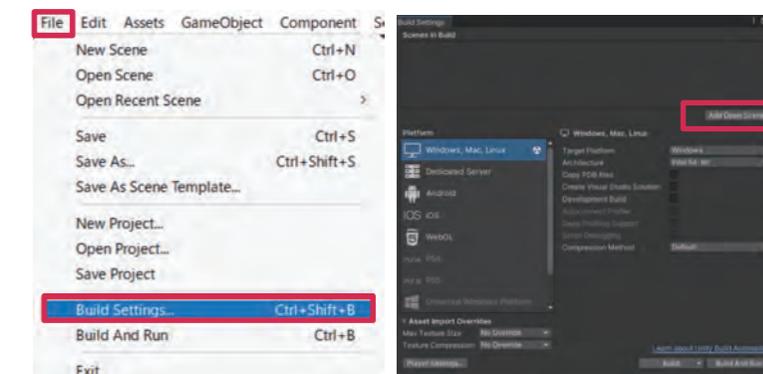


※インポート後、初回のみ表示されます

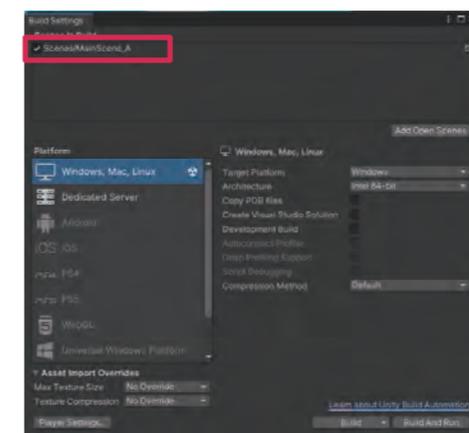
③Game タブに変更すると選択したScenesが表示されます



④File → Build Settings→Add Open Scenes をクリックして選択したスクリーンを追加します



⑤「Add Open Scenes」をクリックして、選択したスクリーンが写っていることを確認します。

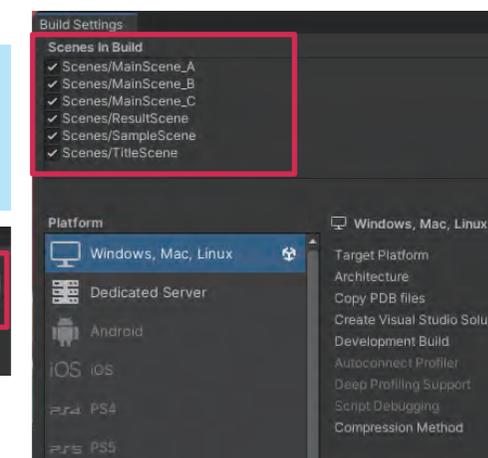


⑥Buildへ残り5つのScenesも同様に追加します



ダブルクリックで対象の Scenes を開いた後、Player Settings で「Add Open Scenes」の追加を5つ分、繰り返します。

⑦6つの「Scenes」が追加されていることを確認します(確認後、設定画面は閉じる)



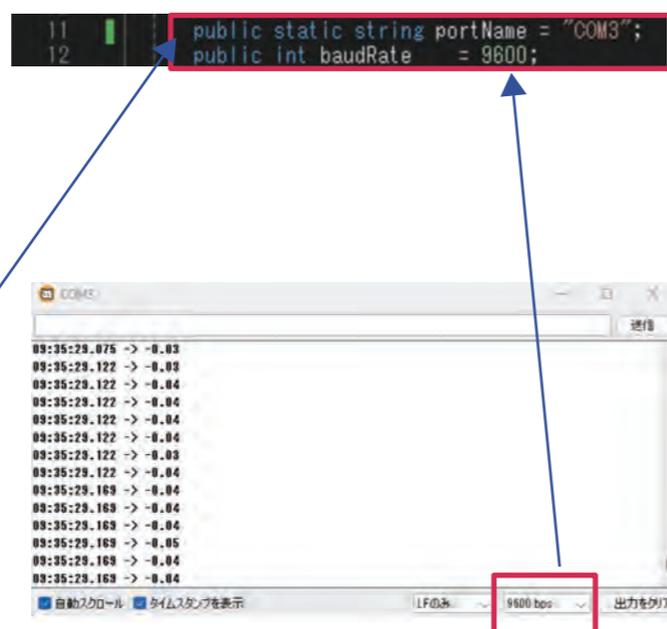
# スクーミーボードと Unity を繋げて、「あつめて! すいかばマン」を動かす

## Unityで「SerialHandler」スクリプトを準備する

①Assets → Scripts で、SerialHandler をダブルクリックして開きます  
中身を、必要に応じて修正しましょう



②ソースコードを修正します  
portNameを、“COM3”のような形式にします  
(ボードを接続しているUSBポートにします。)



## あつめて! すいかばマンをスタートします!

①「TitleScene」が表示されている状態で、「▶」ボタンをクリックします。その後、ゲーム画面の「スタート」をクリックします

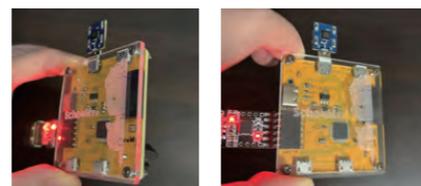


## あつめて! すいかばマンをプレイします!

②すいかが降っているので、スクーミーボードを傾けてキャラクターを動かし、すいかがキャッチしましょう!

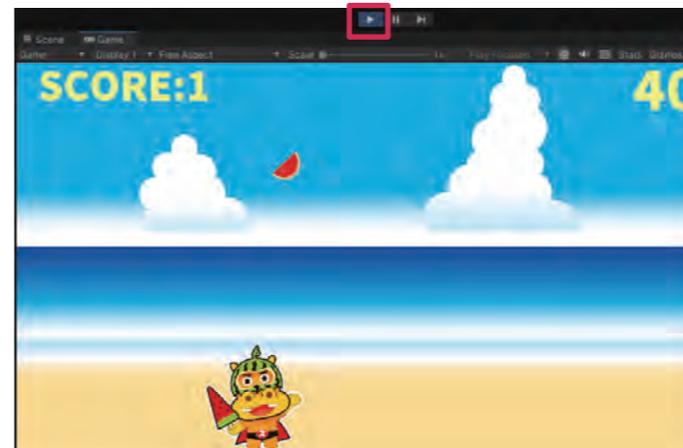


左右にうごくよ!



## あつめて! すいかばマンを停止します!

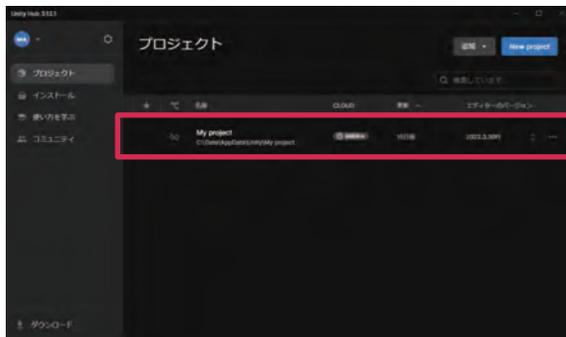
②「▶」ボタンをクリックすると、ゲームが停止しスタート画面に戻る



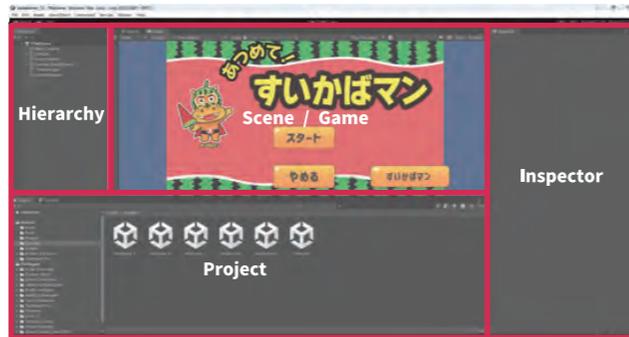
# Unity 画面と役割を理解する

「あつめて! すいかばマン」のプロジェクトを開く

- 1 Unity を起動させ、あつめて! すいかばマンのプロジェクトをクリックします。

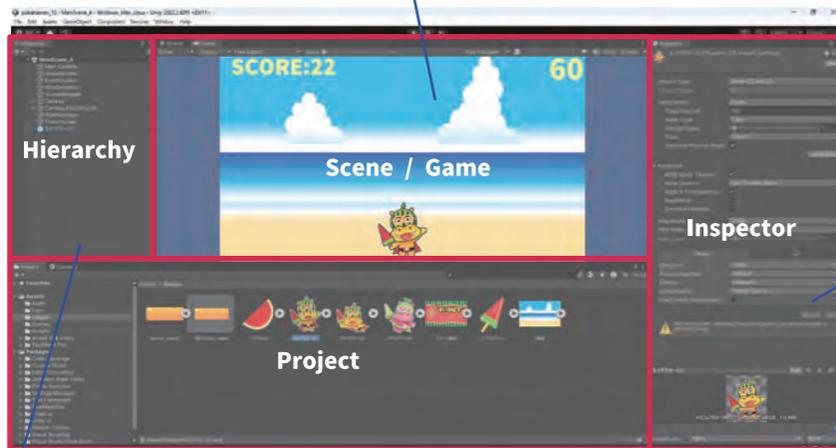


- 2 各ウィンドウの名前です。



## ウィンドウの役割

ゲームの作業空間を表示 ( Scene )  
 実際のゲーム画面を表示 ( Game )



選択したゲームオブジェクトの  
 詳細情報を確認・編集

画像等の素材を追加する場所  
 (C# のスクリプト等はこちら)

ゲームオブジェクトの一覧を表示

- 3 Hierarchy ウィンドウ
  - Scene ビューにあるすべてのオブジェクトが表示される
  - ここでオブジェクト ( 球などの物体 ) を生成できる

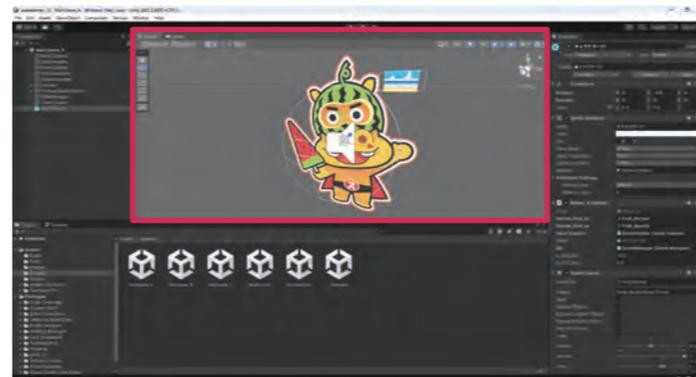


- 4 Scene/Game ウィンドウを切り替える  
 「Scene」でない場合、赤枠の箇所を「Scene」に変更する

Game→  
 Scene に変更



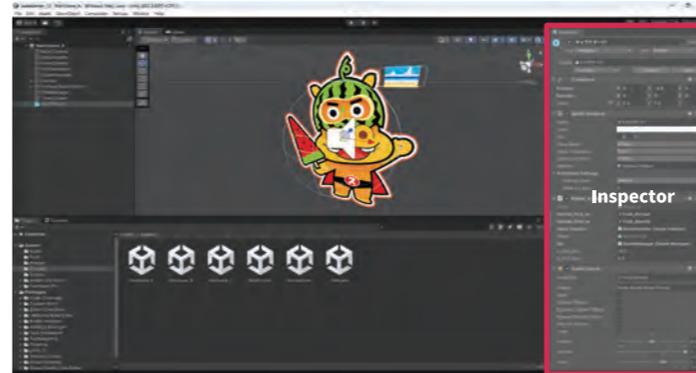
- 5 Scene ウィンドウ
  - Unity のシーン ( Scene ) を編集する画面
  - オブジェクトの位置や大きさ、角度などを変更する



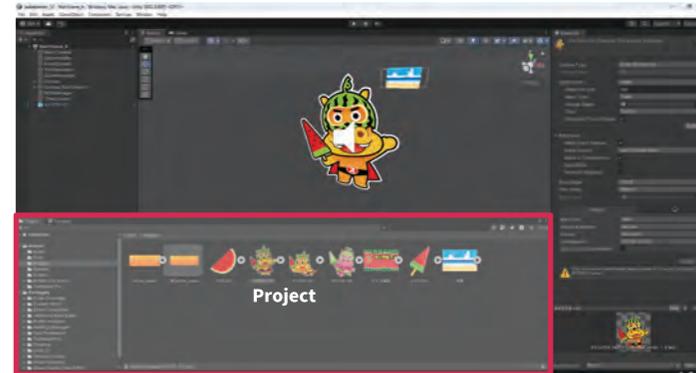
- 6 Game ビュー
  - 現在開発中のゲームを実行して確認できる画面
  - Scene ビューとは異なり、実際にプレイヤーがゲームをプレイしながら、ゲームの動作や不具合をチェックする



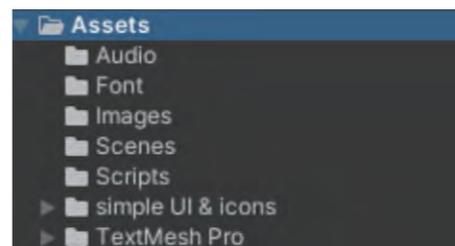
- 7 Inspector ウィンドウ
  - オブジェクトなどの情報を表示、編集する画面
  - 名前、位置、大きさ、衝突判定などを編集できる



- 7 Project ウィンドウ
  - プロジェクトに使うアセット ( キャラクターや建物等のデータ ) を表示する画面
  - ここにアセットを追加してから、Scene ビューで編集する



## Project の Assets でよく使われる各フォルダの説明



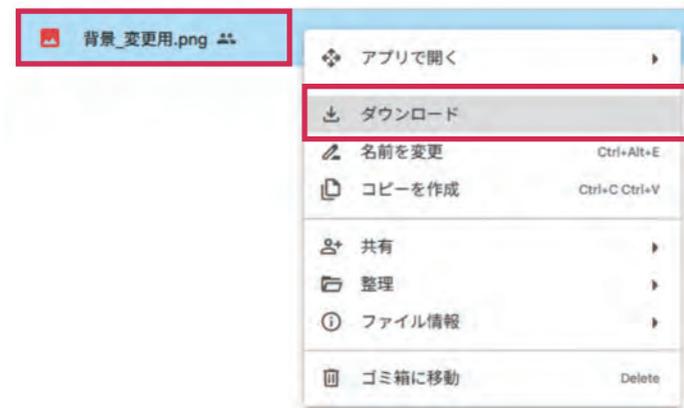
Audio : ゲーム内で使用する音声ファイル ( BGM、効果音など ) を格納  
 Images : 2D スプライトや UI 用の画像など、静止画素材を格納  
 Scenes : ゲームのシーンファイル ( レベルや画面 ) を格納  
 Scripts : C# で記述されたスクリプトを格納

# 背景を変えてみよう！

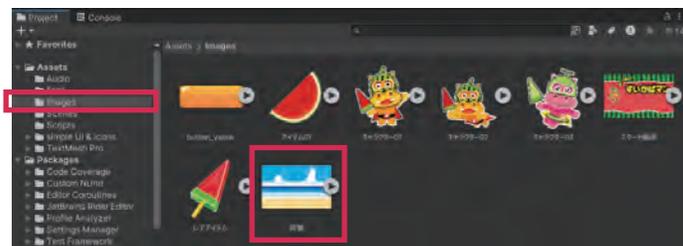
## 事前準備

- 1 変更用の背景をダウンロードします。  
※自分で好きな画像を用意しても OK  
ただし 1280×800 の「png」形式の画像を用意すること。

google ドライブにアクセスし  
「背景\_変更用.png」をダウンロードしましょう。



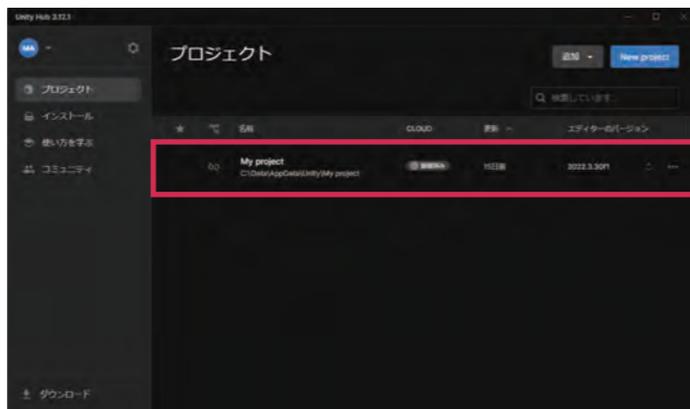
- 3 背景の画像ファイルが保存されている Unity のフォルダを開きます。Assets → images に「背景」のファイルがあることを確認します。



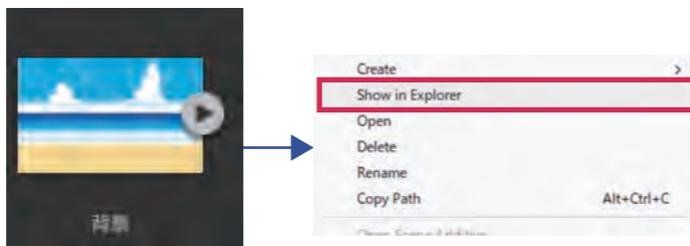
- 5 背景画像のバックアップを作ります。  
画像を選択肢「F2」キーまたは、画像の上で右クリックし「名前の変更」でファイル名を変更します。



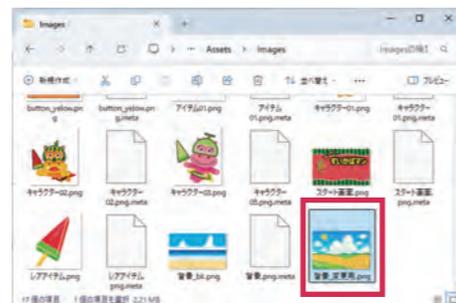
- 2 Unity を起動させて、「あつめて!すいかばマン」のプロジェクトをクリックします。



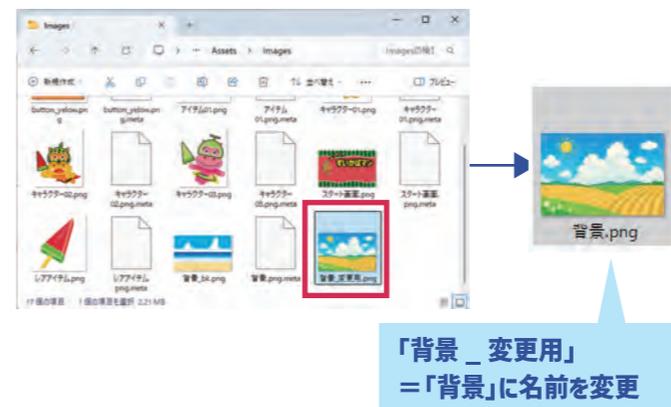
- 4 「背景」の上で右クリック→「Show in Explorer」をクリックして、画像の画像ファイルが保存されているエクスプローラーのフォルダを開きます。



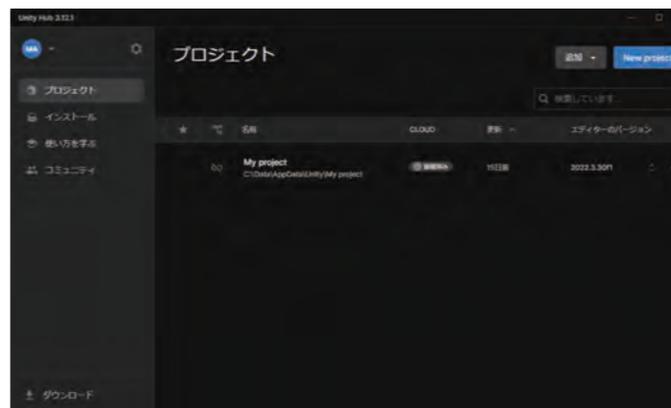
- 6 事前準備でダウンロードした「背景\_変更用.png」を背景と同じフォルダに移動またはコピーします。



- 7 画像を選択して「F2」キーまたは、画像の上で右クリックし「名前の変更」でファイル名を変更します。



- 9 Unity を再度起動させ、先ほどまで作業していた「あつめて!すいかばマン」のプロジェクトを開きます。



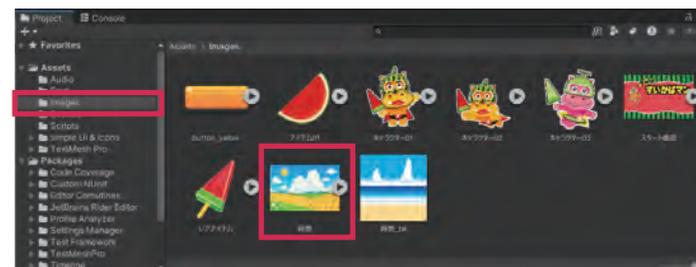
- 11 ゲームを実行してみましょう! 「TitleScene」が表示されている状態で、「▶」ボタンをクリックし、その後、ゲーム画面の「スタート」をクリックします。



- 8 差し替えた画像を読み込ませるために、Unity を再起動させます。右上の「×」ボタンで Unity を終了しましょう。



- 10 Assets → images に変更した「背景」のファイルがあることを確認します。



- 12 海辺のイラストから農場のイラストに、背景画像が変更されたことを確認します。

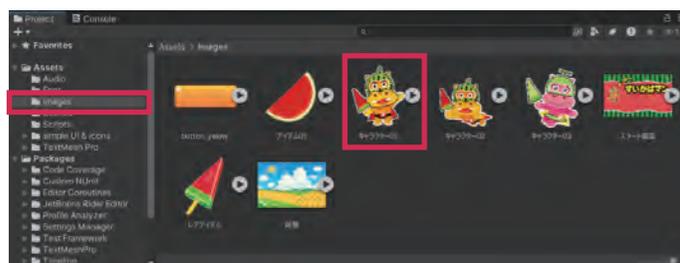


# キャラクターを変えてみよう！

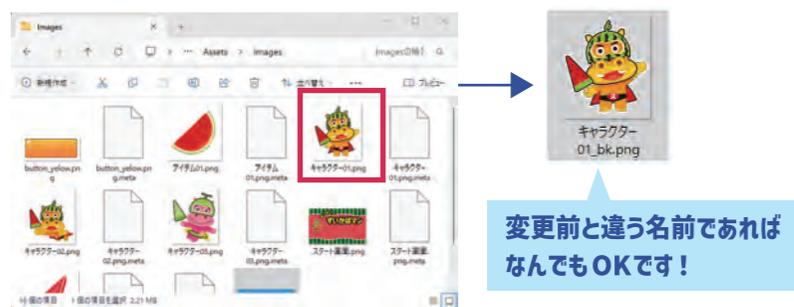
1 変更用のキャラクター画像をダウンロードします。



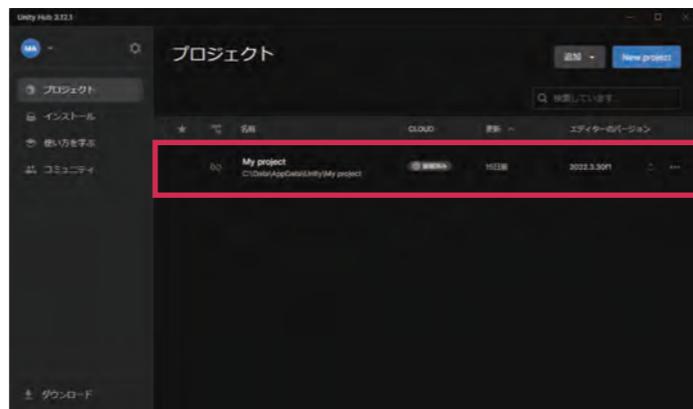
3 背景の画像ファイルが保存されている Unity のフォルダを開きます。Assets → images に「キャラクター」のファイルがあることを確認します。



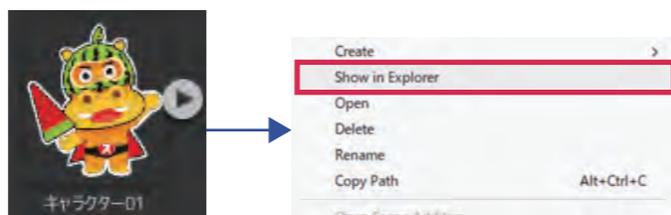
5 キャラクター画像のバックアップを作ります。画像を選択肢「F2」キーまたは、画像の上で右クリックし「名前の変更」でファイル名を変更します。



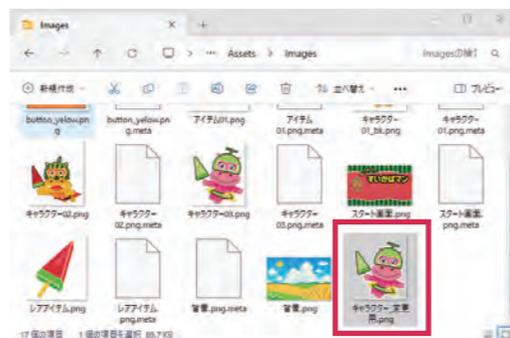
2 Unity を起動させて、「あつめて! すいかばマン」のプロジェクトをクリックします。



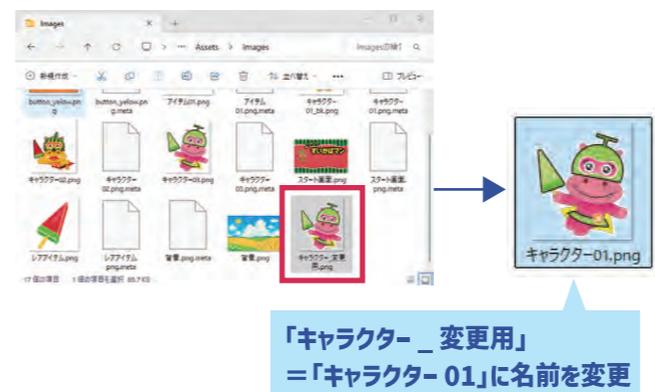
4 「キャラクター」の上で右クリック→「Show in Explorer」をクリックして、画像の画像ファイルが保存されているエクスプローラーのフォルダを開きます。



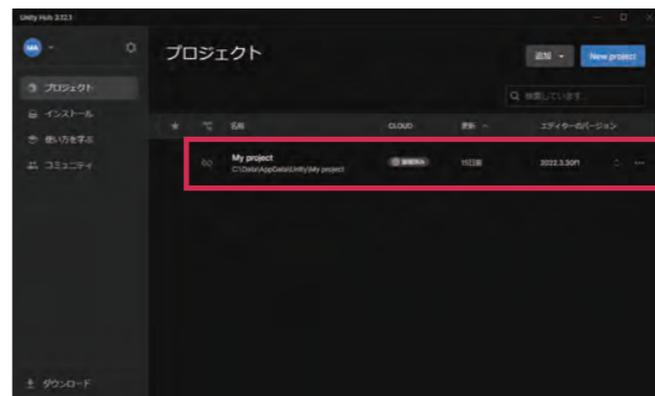
6 事前準備でダウンロードした「キャラクター\_変更用.png」を背景と同じフォルダに移動またはコピーします。



7 画像を選択して「F2」キーまたは、画像の上で右クリックし「名前の変更」でファイル名を変更します。



9 Unity を再度起動させ、先ほどまで作業していた「あつめて! すいかばマン」のプロジェクトを開きます。



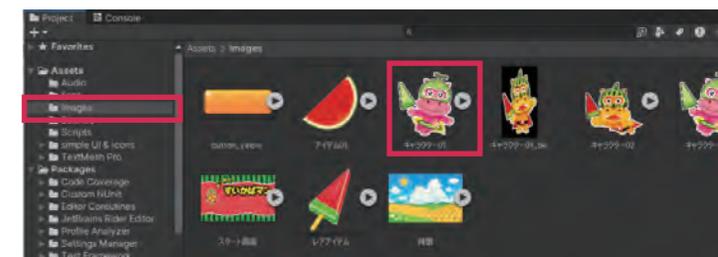
11 ゲームを実行してみましょう! 「TitleScene」が表示されている状態で、「▶」ボタンをクリックし、その後、ゲーム画面の「スタート」をクリックします。



8 差し替えた画像を読み込ませるために、Unity を再起動させます。右上の「×」ボタンで Unity を終了しましょう。



10 Assets → images に変更した「キャラクター」のファイルがあることを確認します。



12 キャラクター画像が変更されたことを確認します。

ぜひ自分で作成・準備したキャラクター画像を設定しましょう!



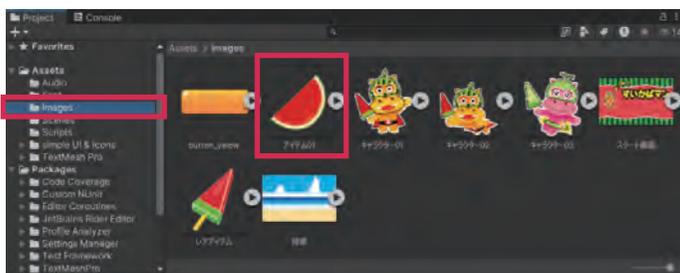
# 落ちてくるものを変えてみよう！

- 1 変更用の画像をダウンロードします。  
※自分で好きな画像を用意してもOK  
ただし画像のサイズを変更前の画像に合わせて作成すること。

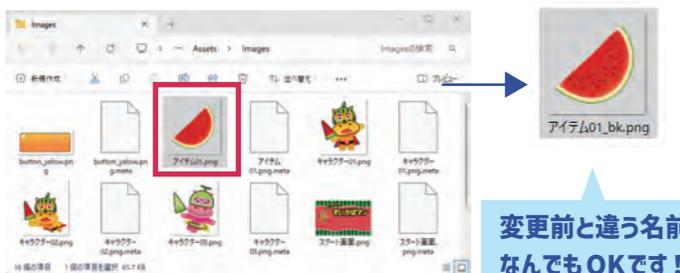
google ドライブにアクセスし  
「すいか変更用 .png」をダウンロードしましょう。



- 3 背景の画像ファイルが保存されている Unity のフォルダを開きます。Assets → images に「アイテム 01」のファイルがあることを確認します。

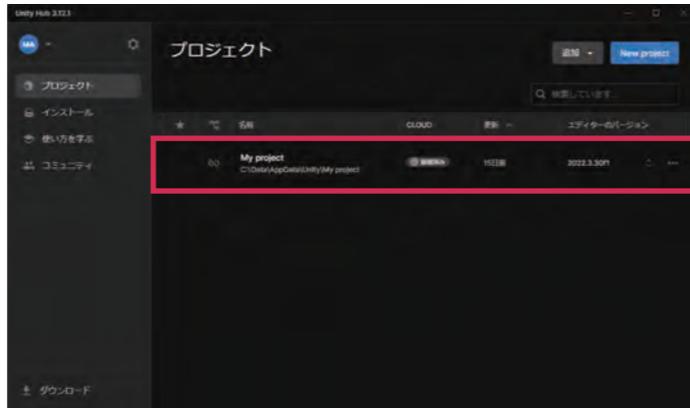


- 5 変更画像のバックアップを作ります。  
画像を選択肢「F2」キーまたは、画像の上で右クリックし「名前の変更」でファイル名を変更します。

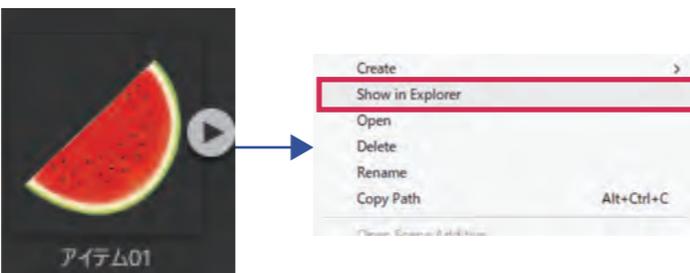


変更前と違う名前であれば  
なんでも OK です！

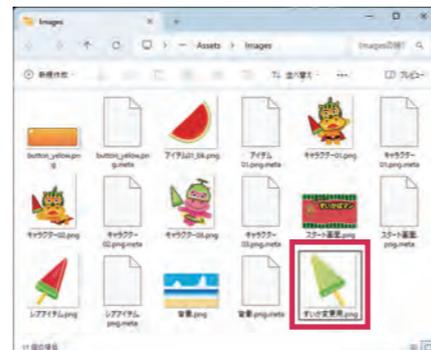
- 2 Unity を起動させて、「あつめて！すいかばマン」のプロジェクトをクリックします。



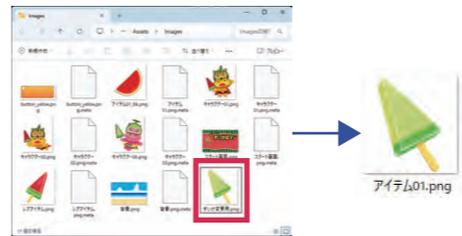
- 4 「アイテム 01」の上で右クリック→「Show in Explorer」をクリックして、画像の画像ファイルが保存されているエクスプローラーのフォルダを開きます。



- 6 事前準備でダウンロードした「すいか変更用 .png」を背景と同じフォルダに移動またはコピーします。



- 7 画像を選択して「F2」キーまたは、画像の上で右クリックし「名前の変更」でファイル名を変更します。

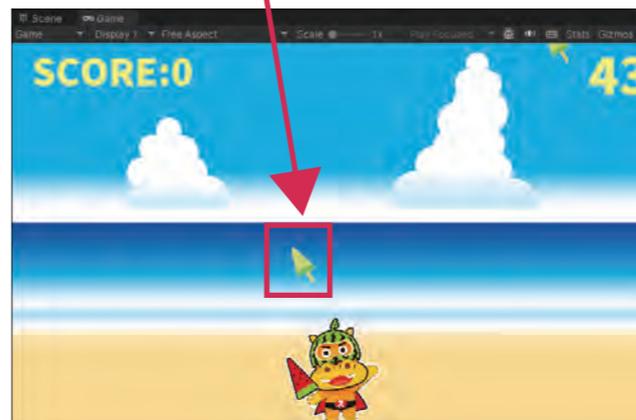


「すいか変更用 .png」  
＝「アイテム 01」に名前を変更

- 10 Assets → images に変更した「アイテム 01」のファイルがあることを確認します。



- 12 スイカからメロンバーに画像が変更されたことを確認します。

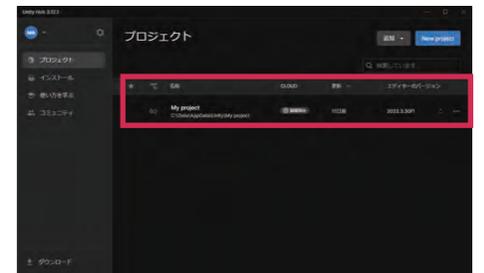


ゲームの動きには関係ありませんが、  
名前の不一致が気になる場合は変更してください  
画像を選択し「F2」キーまたは、画像の上で右クリックし「名前の変更」または「Rename」で、ファイル名を変更します。

- 8 差し替えた画像を読み込ませるために、Unity を再起動させます。右上の×ボタンで Unity を終了しましょう。



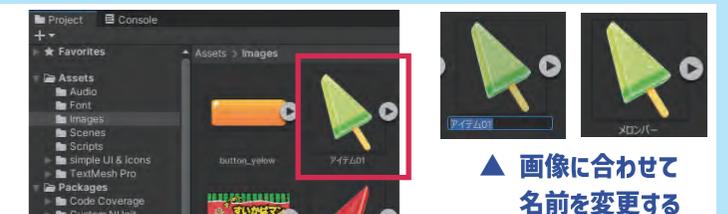
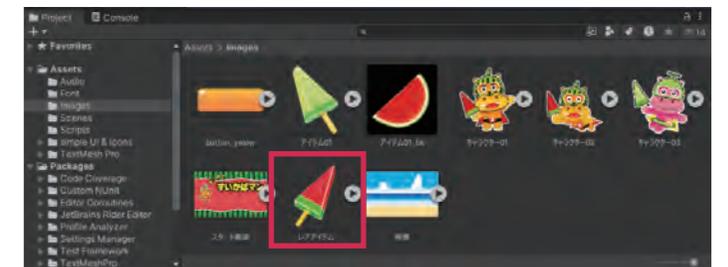
- 9 Unity を再度起動させ、先ほどまで作業していた「あつめて！すいかばマン」のプロジェクトを開きます。



- 11 ゲームを実行してみましょう！「TitleScene」が表示されている状態で、「▶」ボタンをクリックし、その後、ゲーム画面の「スタート」をクリックします。



- 13 他のアイテムも変更してみましょう！



▲ 画像に合わせて  
名前を変更する

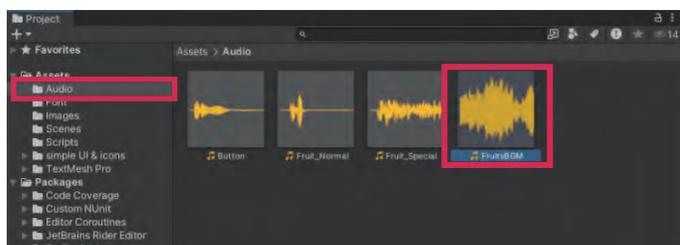
## BGMを変えてみよう！

- 1 変更用の BGM をダウンロードします。  
※自分で好きな BGM を用意しても OK  
ただしイベント等で使用する場合は著作権にご注意ください。

google ドライブにアクセスし  
「変更用 BGM.mp3」をダウンロードしましょう。



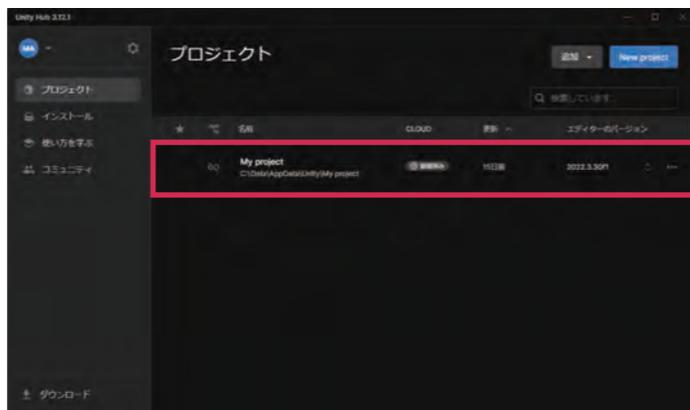
- 3 BGM ファイルが保存されている Unity のフォルダを開きます。  
Assets → Audio に「FruitsBGM」のファイルがあることを確認します。



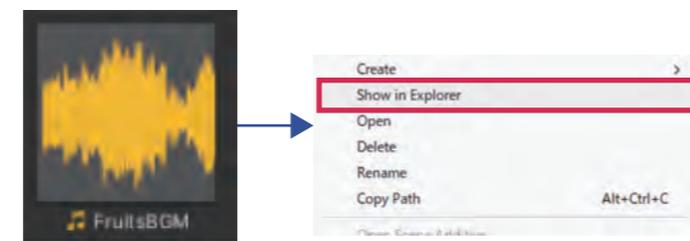
- 5 BGM のバックアップを作ります。  
画像を選択肢「F2」キーまたは、画像の上で右クリックし  
「名前の変更」でファイル名を変更します。



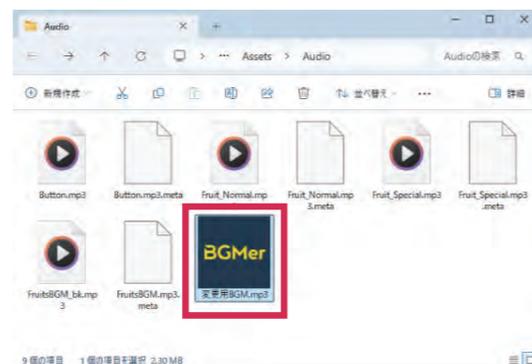
- 2 Unity を起動させて、「あつめて!すいかばマン」のプロジェクトをクリックします。



- 4 「FruitsBGM .mp3」の上で右クリック→「Show in Explorer」をクリックして、BGM ファイルが保存されているエクスプローラーのフォルダを開きます。



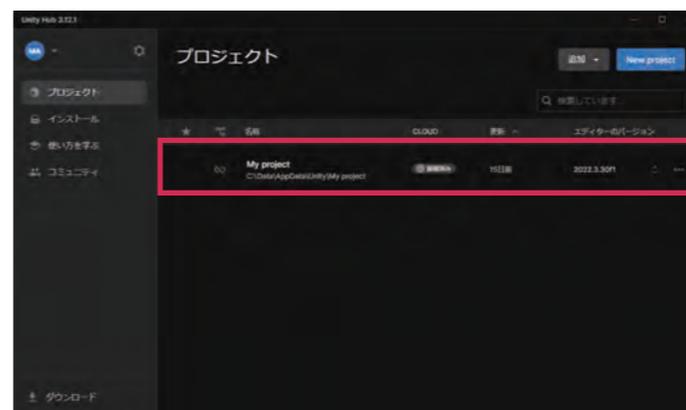
- 6 事前準備でダウンロードした「BGM\_変更用 .mp3」を BGM と同じフォルダに移動またはコピーします。



- 7 BGMを選択して「F2」キーまたは、画像の上で右クリックし「名前の変更」でファイル名を変更します。



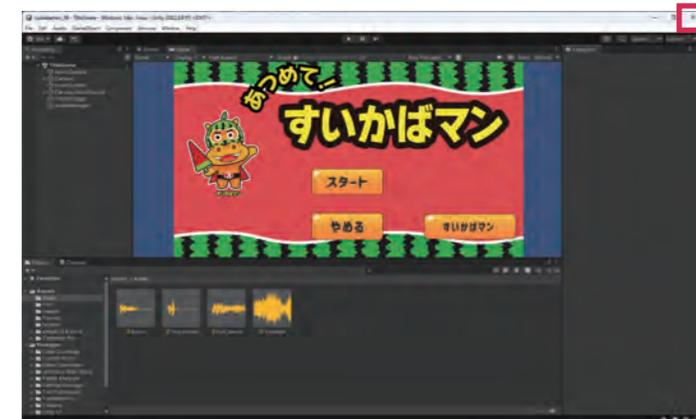
- 9 Unity を再度起動させ、先ほどまで作業していたあつめて!すいかばマンのプロジェクトを開きます。



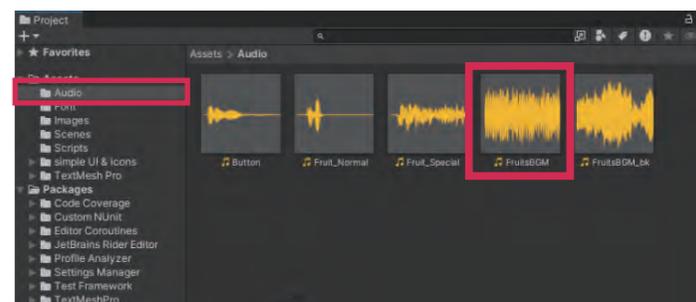
- 11 ゲームを実行してみましょう! 「TitleScene」が表示されている状態で、「▶」ボタンをクリックし、その後、ゲーム画面の「スタート」をクリックします。



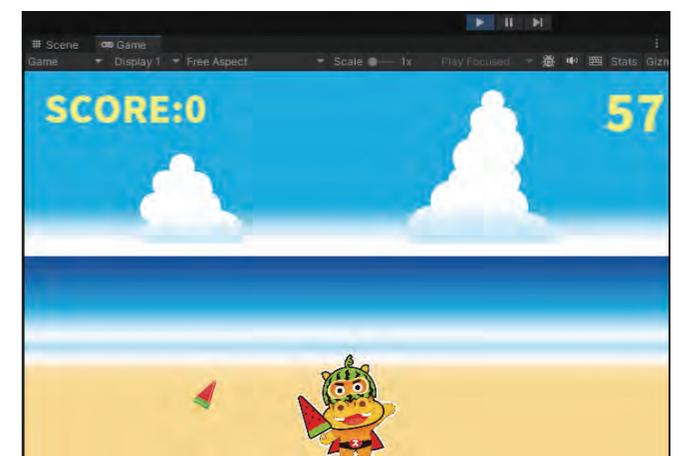
- 8 差し替えた BGMを読み込ませるために、Unity を再起動させます。右上の「×」ボタンで Unity を終了しましょう。



- 10 Assets → Audio に変更した「FruitsBGM.mp3」のファイルがあることを確認します。



- 12 ゲーム中の BGM が変更されたことを確認しましょう。



# スタート画面を変えてみよう！

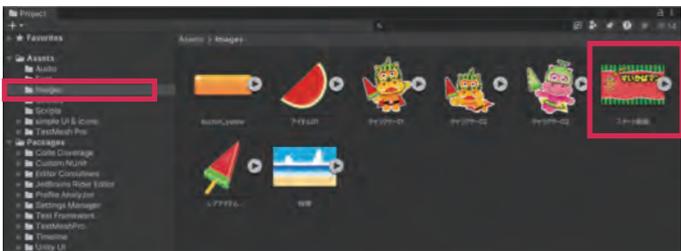
## 事前準備

- 1 変更用の背景をダウンロードします。  
※自分で好きな画像を用意しても OK  
ただし 1280×800 の「png」形式の画像を用意すること

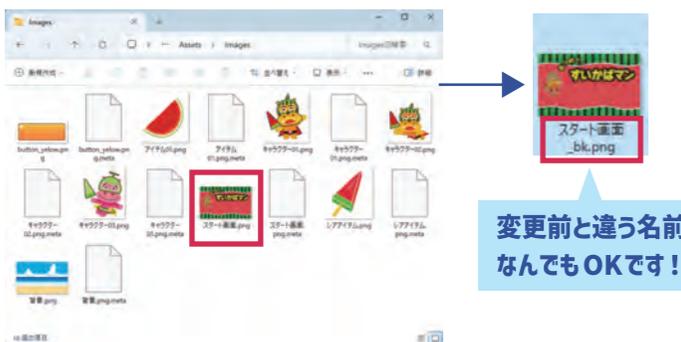
google ドライブにアクセスし「スタート画面\_背景\_変更用.png」をダウンロードしましょう。



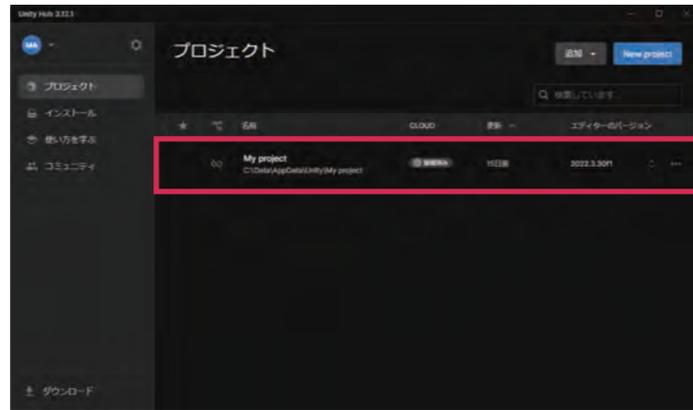
- 3 スタート画面の画像ファイルが保存されている unity のフォルダを開きます。Assets → images に「スタート画面」のファイルがあることを確認します。



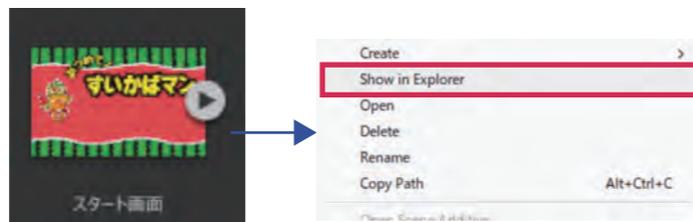
- 5 画像のバックアップを作ります。  
画像を選択肢「F2」キーまたは、画像の上で右クリックし「名前の変更」でファイル名を変更します。



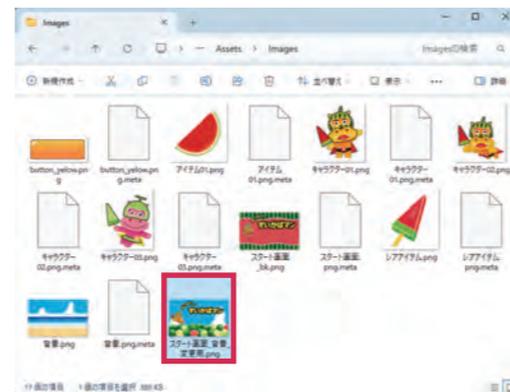
- 2 Unity を起動させて、「あつめて!すいかばマン」のプロジェクトをクリックします。



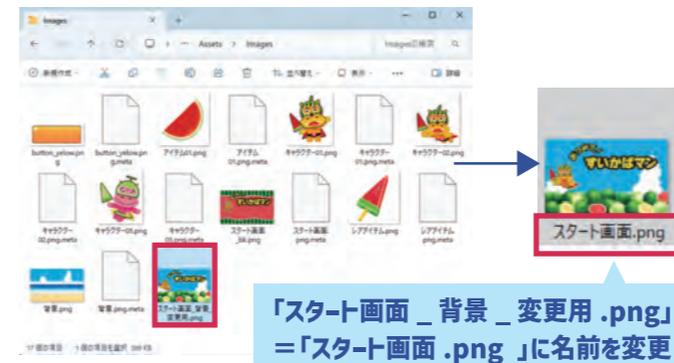
- 4 スタート画面の画像ファイルが保存されているエクスプローラーのフォルダを開きます。「スタート画面」の上で右クリック→「Show in Explorer」をクリックします。



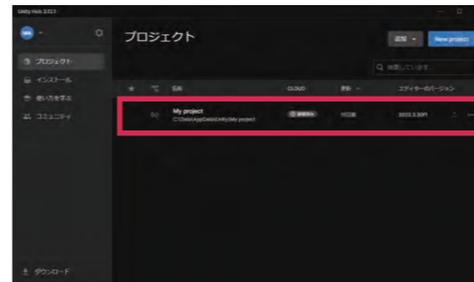
- 6 事前準備でダウンロードした「スタート画面\_背景\_変更用.png」をスタート画面の画像と同じフォルダに移動またはコピーします。



- 7 画像を選択して「F2」キーまたは、画像の上で右クリックし「名前の変更」でファイル名を変更します。



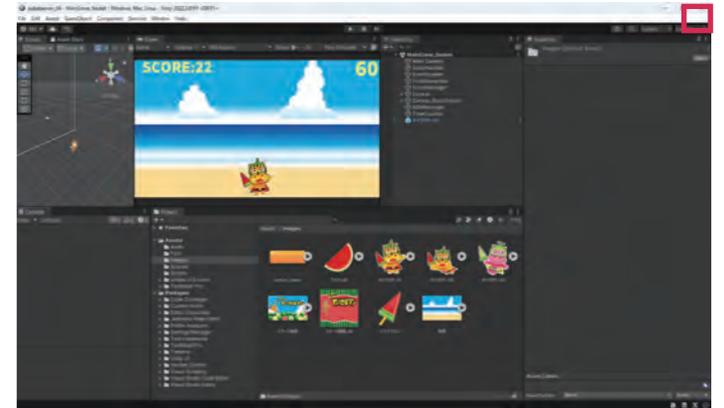
- 9 Unity を再度起動させ、先ほどまで作業していた「あつめて!すいかばマン」のプロジェクトを開きます。



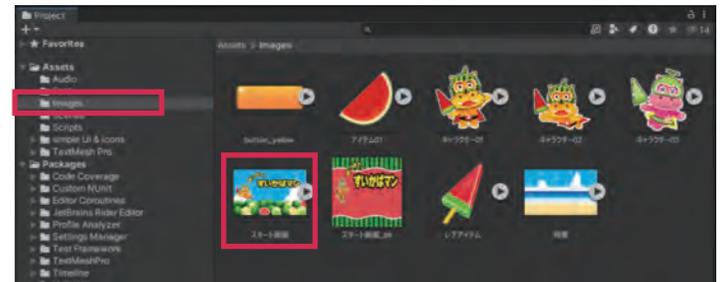
- 11 変更されたスタート画面を確認します。「TitleScene」をダブルクリックして、スタート画面の画像が変更されたことを確認します。



- 8 差し替えた画像を読み込ませるために、Unity を再起動させます。右上の「×」ボタンで Unity を終了しましょう。

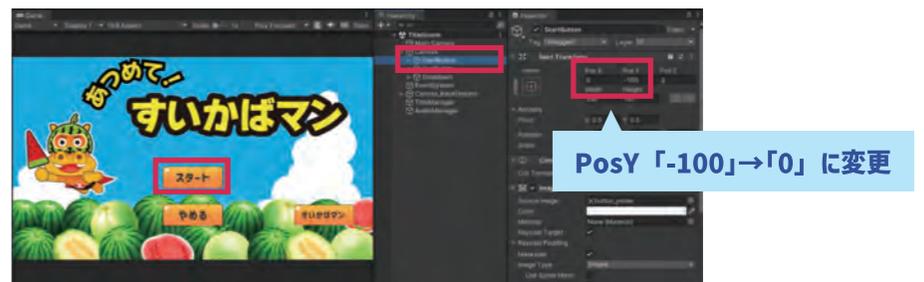


- 10 Assets → imagesに変更した「スタート画面」のファイルがあることを確認します。

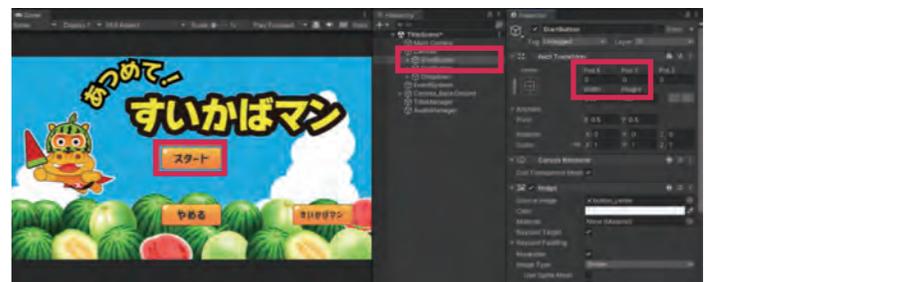


## ボタンの一を変更する

- 12 Hierarchy → Canvas → StartButton をクリックして、Inspector の赤枠を変更することで、ボタンの位置を変更できます。

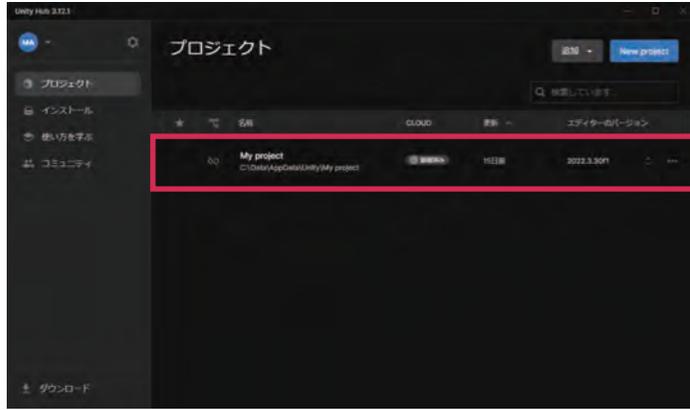


- 13 Game 画面のスタートボタンの位置が変わっていることが確認しましょう。同様に他のボタンの位置も変更できます。「QuitButton」→ やめるボタン「DropButton」→ キャラクター変更ボタン



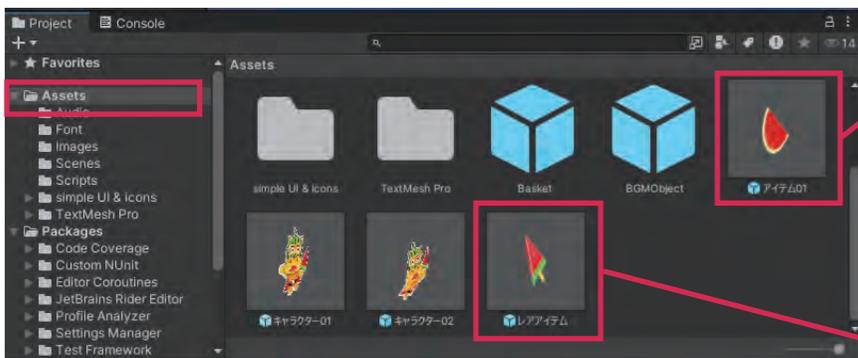
# 得点を変えてみよう!

1 Unityを起動させて、「あつめて!すいかばマン」のプロジェクトをクリックします。



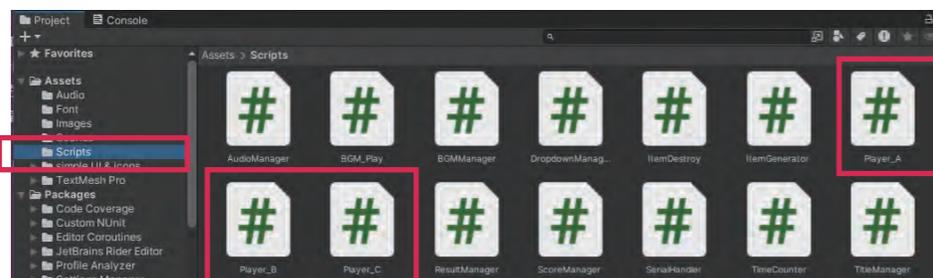
3 アイテムの得点を管理しているファイルを確認します。Assets → 「アイテム 01」のファイルがあることを確認します。

Assetsの対象を選択して「Inspector」の「Tag」を確認します  
アイテム01は「NormalFruit」  
レアアイテムは「SpecialFruit」が設定されている



4 Assets → Scripts に以下のファイルがあることを確認します。

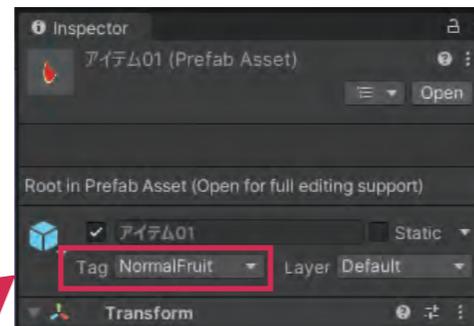
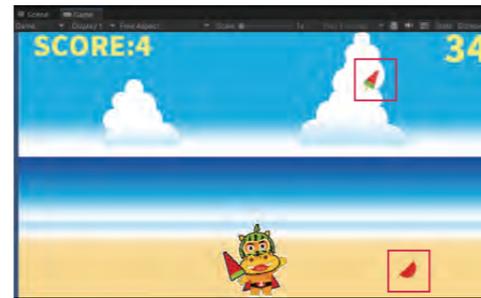
Player\_A.cs  
Player\_B.cs  
Player\_C.cs



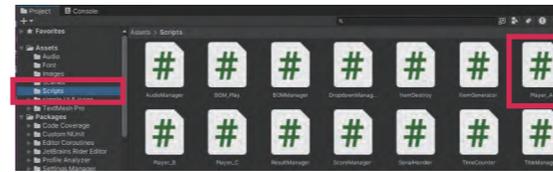
2 得点の仕組みを整理しましょう。アイテムをキャッチすると得点 (SCORE) が増えます。このキャッチした時のアイテムの得点を変更します。

【変更前】  
スイカ → 1点  
スイカパー → 3点

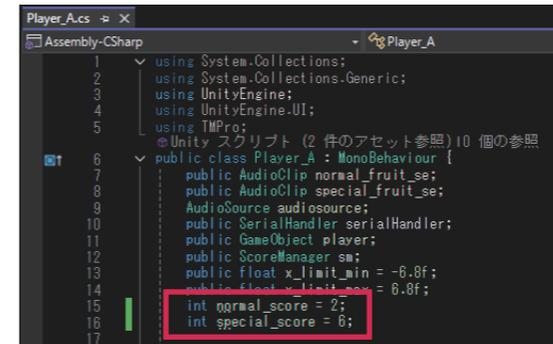
【変更後】  
スイカ → 2点  
スイカパー → 6点



5 アイテムの得点を管理しているファイルを開きます。「Player\_A.cs」をダブルクリックして編集画面を開きます。



7 ソースコードを保存します。「Ctrl + S」または、ファイルを閉じて「保存」ボタンをクリックしましょう。

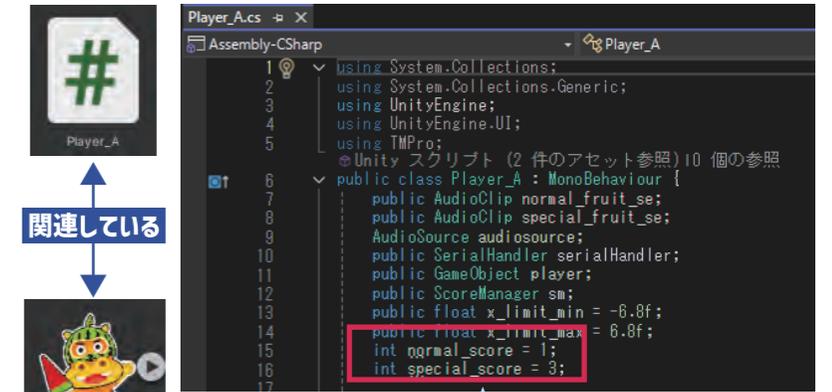


修正後

9 得点の変更を確認してみましょう。キャッチした時のアイテムの得点が、変更後の得点になっていることを確認してみてください。



6 ソースコードを修正 (アイテムの得点を変更) します。「Player\_A.cs」の得点を管理している箇所を、以下のように修正しましょう。



以下のように変更します  
int normal\_score = 2;  
int special\_score = 6;

8 ゲームを実行してみましょう! 「TitleScene」が表示されている状態で、「▶」ボタンをクリックし、その後、ゲーム画面の「スタート」をクリックします。



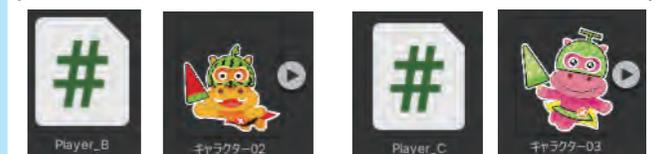
修正済



10 残りのソースコードも修正します。

Assets → Scripts の以下のファイルに対して  
「Player\_A.cs」と同じように得点のソースコードを修正する  
・ Player\_B.cs  
・ Player\_C.cs

未修正



# 当たり判定を変えてみよう！

## 当たり判定の位置の変更とは

- 1 キャラクターを変更すると、キャッチする位置が変わるので当たり判定の位置を変更します！

当たり判定の位置が左下

当たり判定の位置が右上



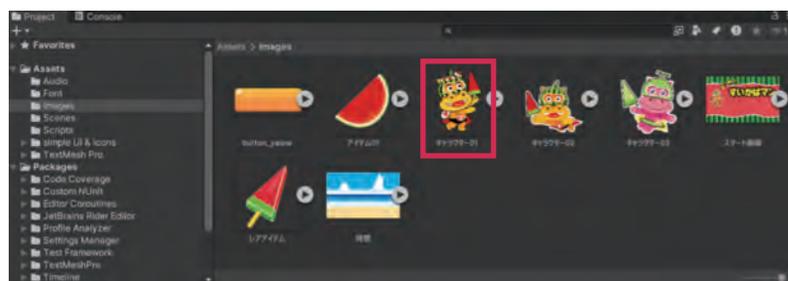
▲変更前



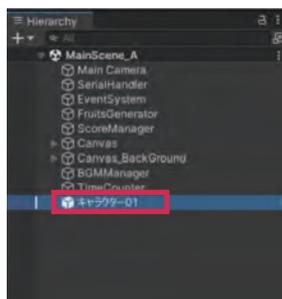
▲変更後

## 準備：キャラクターの画像を変える

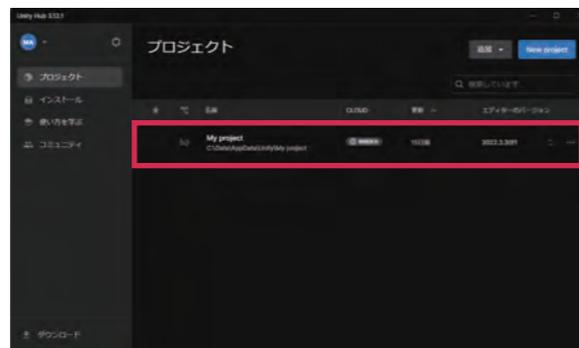
- 3 「キャラクターを変えてみよう！」の手順を参考にキャラクターを変更します。(自分で用意したもの、もしくは、google ドライブにアクセスし「キャラクター\_01\_当たり判定変更用.png」をダウンロードした画像を使う)



- 5 当たり判定の設定項目を開きます。  
Hierarchy → キャラクター\_01 をクリックします。

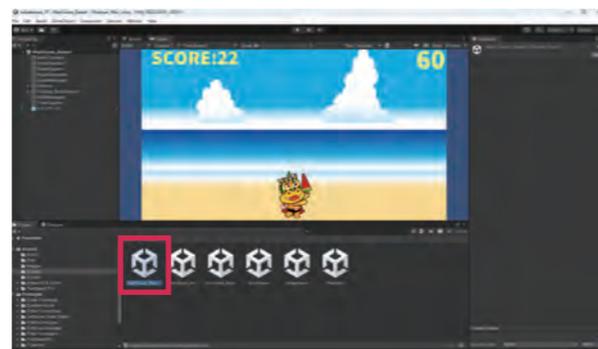


- 2 Unity を起動させ、「あつめて! すいかばマン」のプロジェクトをクリックします。



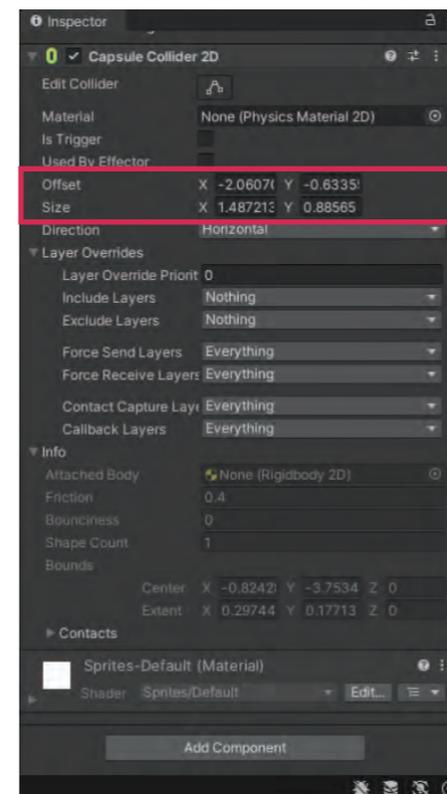
- 4 変更したキャラクターが登場する Scenes を開きます。

Assets → Scenes → MainScene\_A をダブルクリックする  
※変更したキャラクターによって、MainScene\_A ~ C それぞれを修正する



- 6 当たり判定の設定項目を確認します。

Inspector の「Capsule Collider 2D」の項目の「Offset」と「Size」の値を変更することで当たり判定の位置や大きさを変更できます。

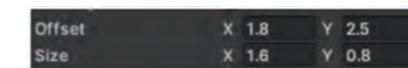


- 7 当たり判定の位置の変更をします。  
Scene 画面を開くと当たり判定の範囲を確認できるので、範囲を見ながら位置と大きさを変更しましょう。



▶変更前

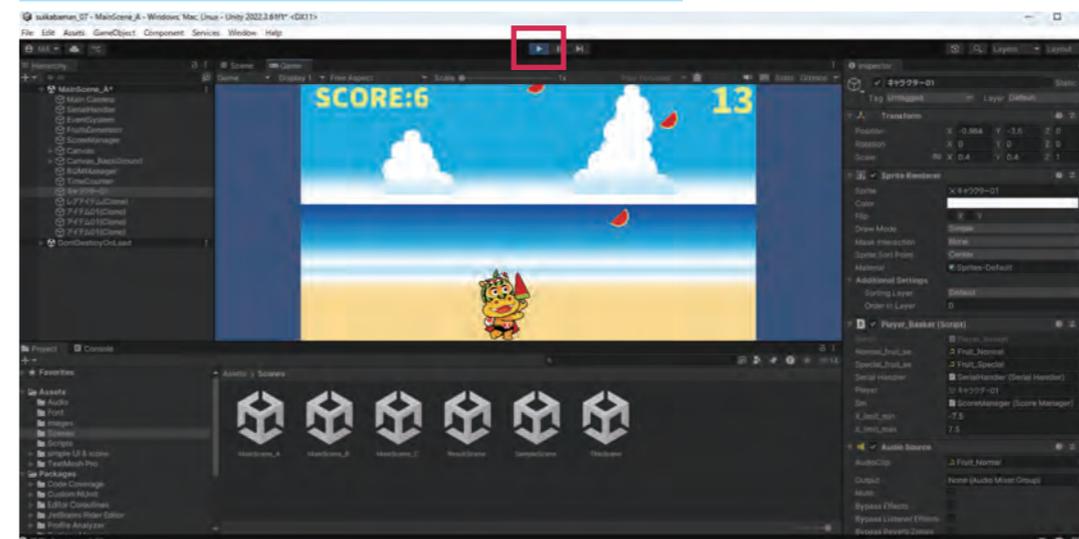
Offset は、Collider (当たり判定の範囲) の中心位置を、オブジェクトの中心からどれだけズラすかを指定する値



▶変更後

- 8 ゲームを実行させ、当たり判定の位置の変更を確認します。

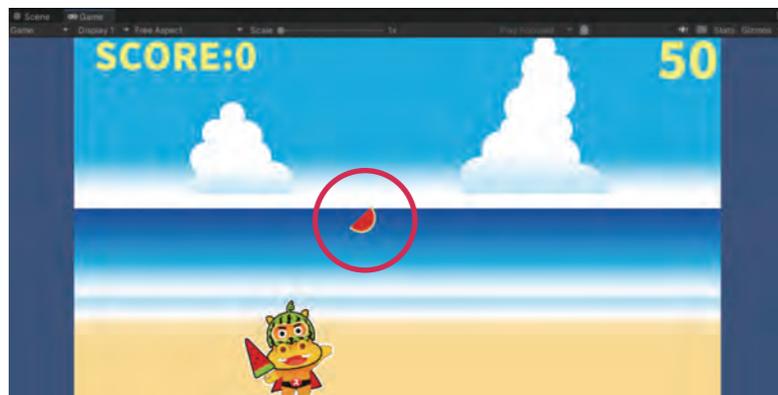
Game 画面を開き、「▶」ボタンをクリックして、想定した位置でキャッチできるか確認します。



# 落ちてくるアイテムの時間間隔を変えてみよう！

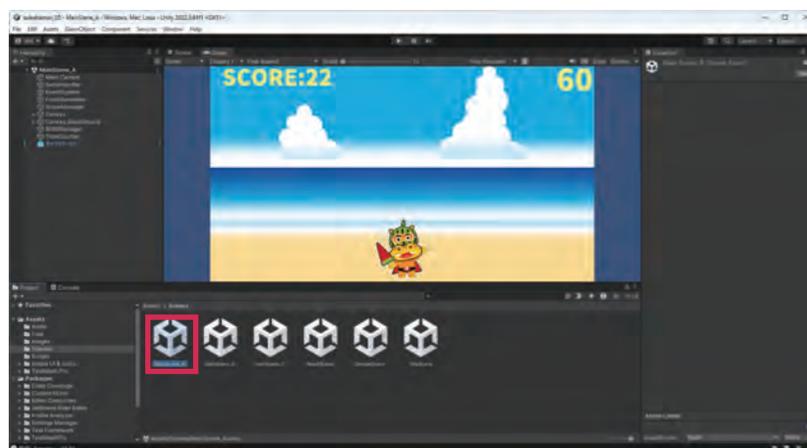
## 落ちてくるアイテムの時間間隔の変更とは

- 1 しばらく待たないと落ちてこないアイテムが頻繁に落ちてくるように変更します。

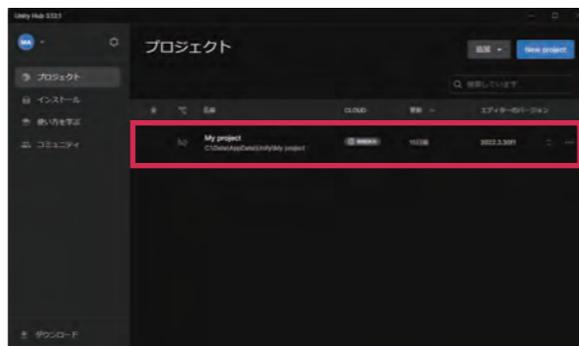


- 3 変更したキャラクターが登場する Scenes を開きます。

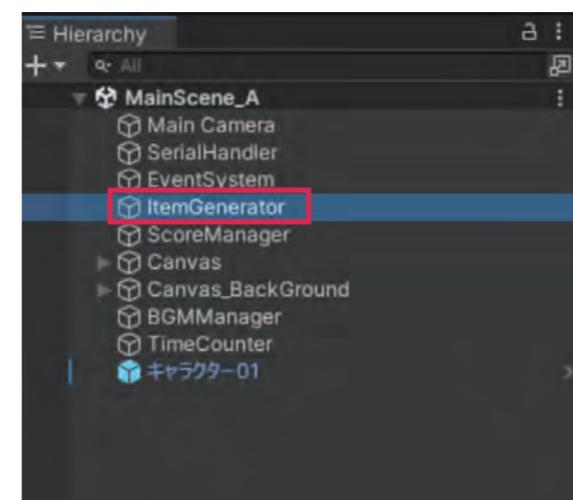
Assets → Scenes → MainScene\_A をダブルクリックする  
※変更したい Scenes によって、MainScene\_A ~ C それぞれを修正します。



- 2 Unity を起動させ、「あつめて！すいかばマン」のプロジェクトをクリックします。

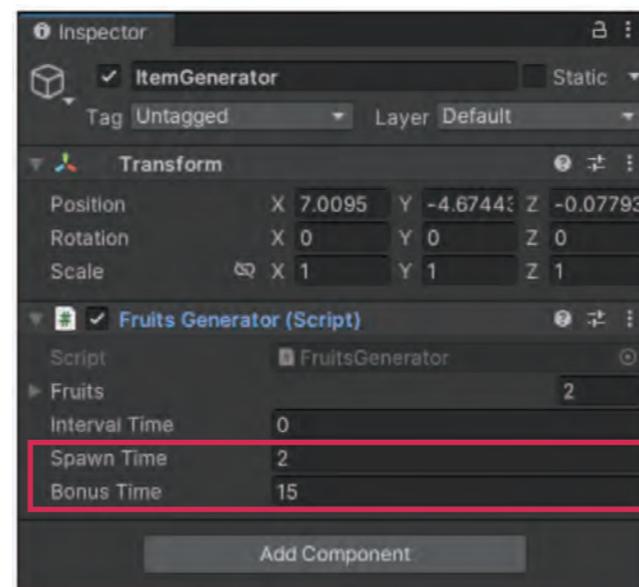


- 4 落ちてくるアイテムの設定項目を開きます。  
Hierarchy → ItemGenerator をクリックします。

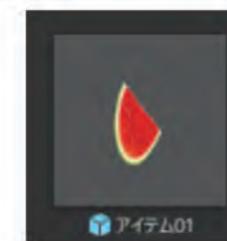
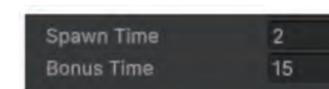


- 5 落ちてくるアイテムの設定項目を確認します。

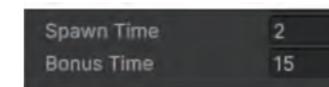
Inspector の「Fruits Generator (Script)」の項目の「Spawn Time」と「Bonus Time」の値を変更することでアイテムの作成タイミングを変更できます。



- 6 Spawn Time はアイテムを一定間隔で出現させる時間の目安を設定します。Bonus Time はレアアイテムなど特別なものを出す時間の目安を設定します。



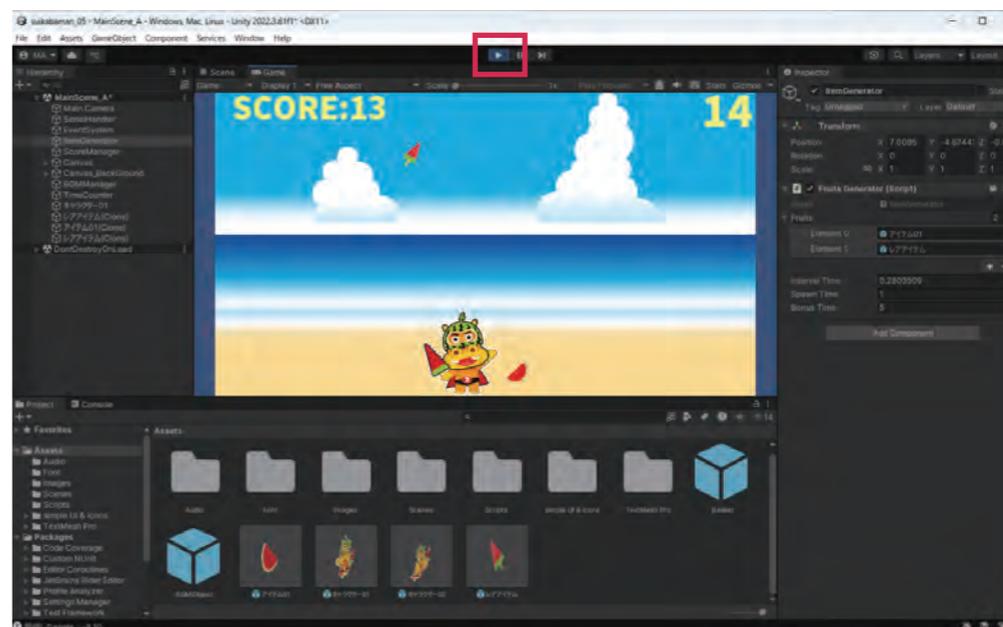
アイテムを  
2秒ごとに出現させる  
↓  
1秒に変更するとアイテムの  
出現が多くなる



レアアイテムを15秒ごとに  
出現させる  
↓  
5秒に変更すると  
レアアイテムが頻繁に  
出現する

- 7 ゲームを実行させ、落ちてくるアイテムの時間間隔の変更を確認しましょう。

Game 画面を開き、「▶」ボタンをクリックし、落ちてくるアイテムの時間間隔が変わっているか確認します。



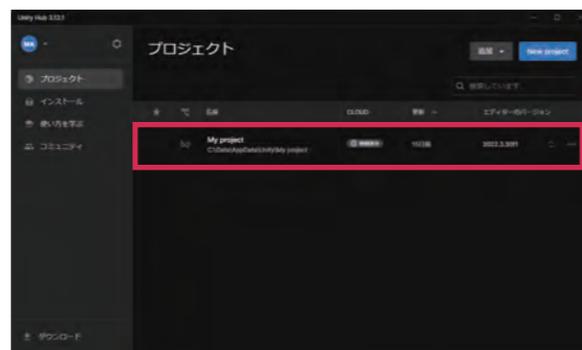
# アイテム出現位置の確率（ランダム）を変えてみよう！

## アイテム出現位置の確率（ランダム）の変更とは

1 アイテムは左右のランダムな位置から落ちてくるようになっています。難易度を変更するために、アイテム出現位置の確率（ランダム）を変更しましょう。

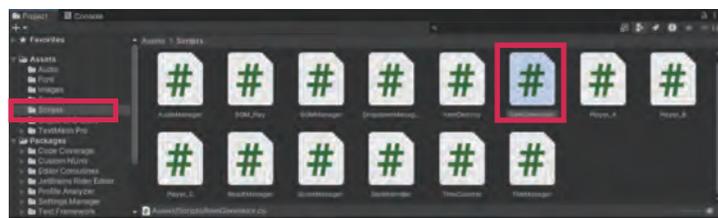


2 Unity を起動させ、「あつめて！すいかばマン」のプロジェクトをクリックします。



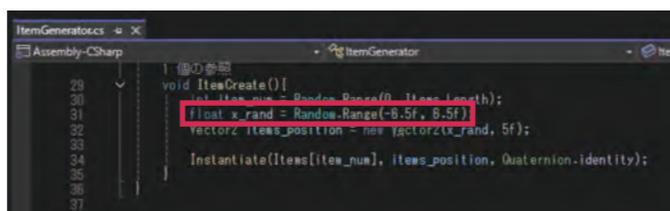
3 変更する Scripts ファイルを開きます。

Assets → Scripts → ItemGenerator を開きます。



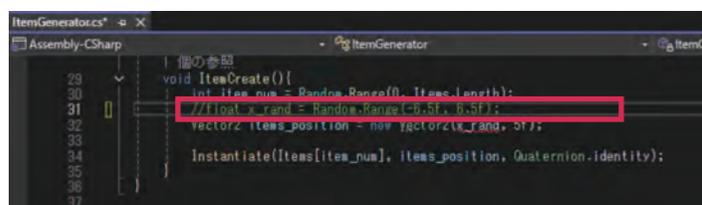
4 変更する Scripts ファイルの内容を確認します。

ItemGenerator 31 行目に横方向 X 軸の「-6.5f ~ 6.5f」の範囲でランダムにアイテムが出現するという処理が設定されている。  
( f : float 型の数値という意味)



5 Scripts ファイルの内容を変更します。

31 行目の頭に「//」を付けて元の処理をコメントアウトします。

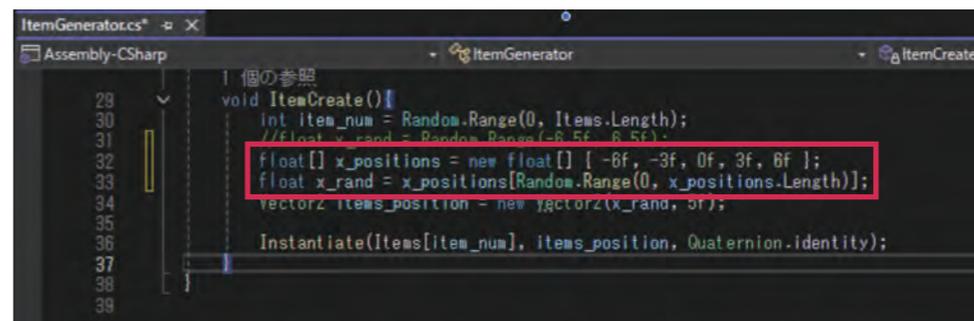


## パターン① 決まった位置からの落下

6 Scripts ファイルの内容を変更します。変更したら、ゲームを実行しましょう。

落下位置を「決まった位置」からにします。32行目に以下を追加し、ファイルを保存しましょう。

```
float[] x_positions = new float[] { -6f, -3f, 0f, 3f, 6f };
float x_rand = x_positions[Random.Range(0, x_positions.Length)];
```

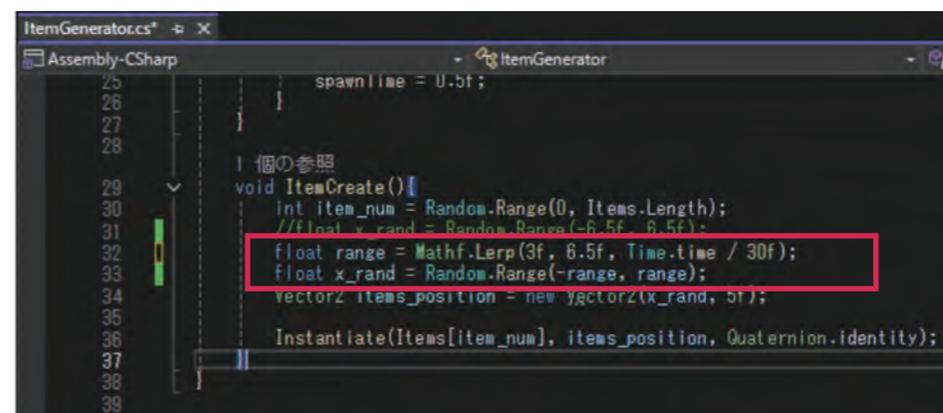


## パターン② 時間経過で範囲が広がる

7 Scripts ファイルの内容を変更します。変更したら、ゲームを実行しましょう。

時間経過で範囲が広がる (30 秒で最大範囲) 32 行目に以下を追加し、ファイルを保存します。

```
float range = Mathf.Lerp(3f, 6.5f, Time.time / 30f);
float x_rand = Random.Range(-range, range);
```



パターン①②それぞれ変更をしたあとは、ゲームを実行して変更が反映されているか確認しましょう。

8 ゲームを実行させ、落ちてくるアイテムの変更を確認します。

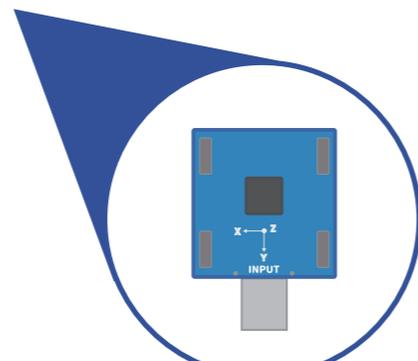
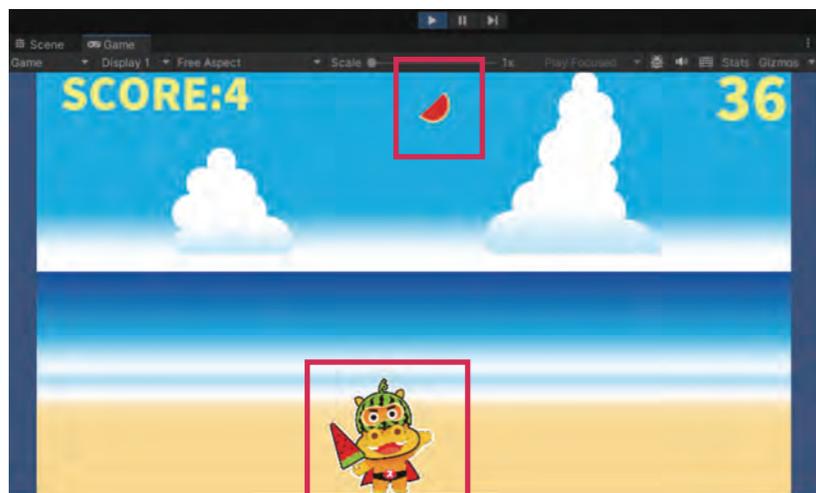
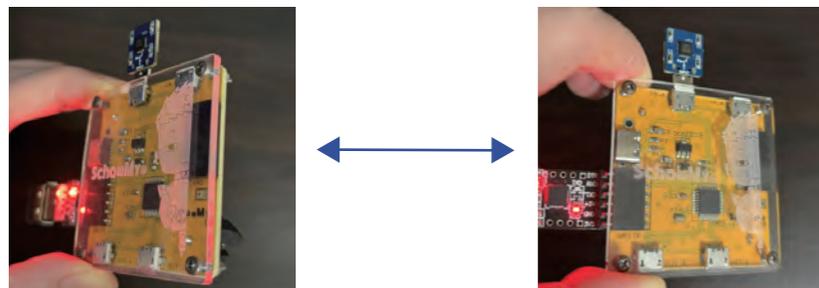


Game 画面を開き、「▶」ボタンをクリックし、落ちてくるアイテムの出現位置が変わっているか確認する

# コネクタのプログラムを変えてみよう！（方向軸の変更）①

## コネクタのプログラムと方向軸について

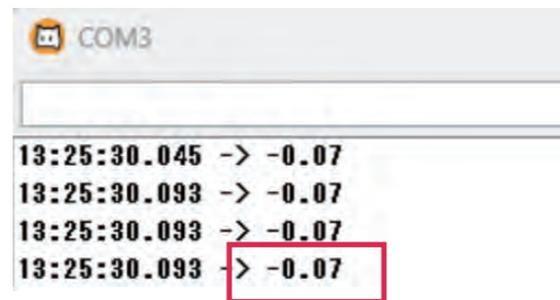
1 手順書で作成したゲームはスクリーンボードを左右に傾ける（Y軸を変更させる）ことで、キャラクターを動かして、アイテムをキャッチさせます。



加速度センサーをよく見ると方向軸が記載されています

## コネクタのプログラム

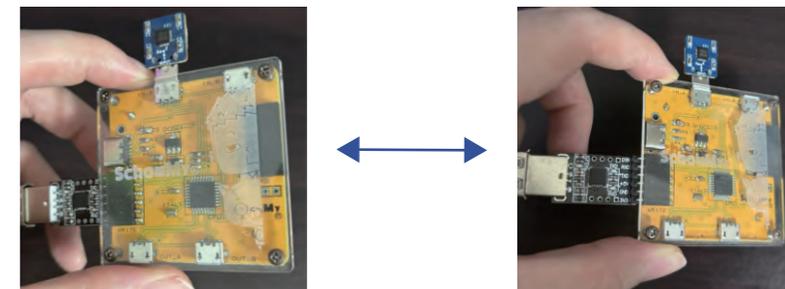
2 シリアルモニタに「y」の値が表示されるブロックコードを使っています。



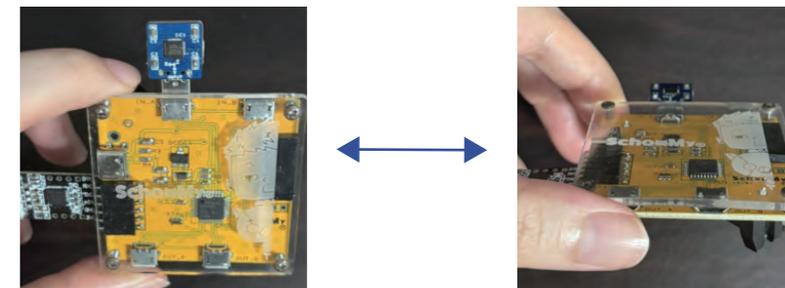
## コネクタのプログラムの方向軸（Y軸、X軸、Z軸）について

3 変更する Scripts ファイルを開きます。

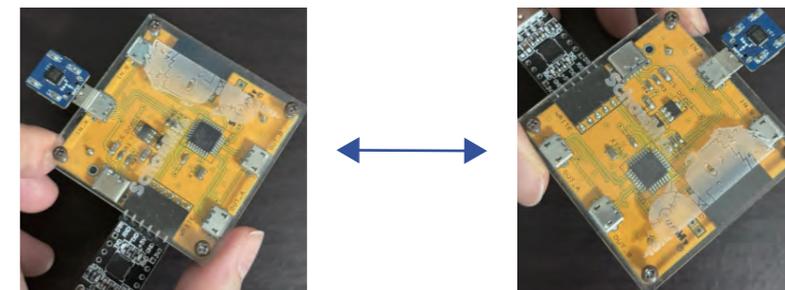
Y軸回転：右左に傾ける  
そのままコントローラを傾けるタイプに向いている



X軸回転：上下に傾ける  
釣竿や棒、刀など、振り下ろすタイプに向いている



Z軸回転：左右に回転させる  
車のハンドルや、船の舵輪、ダイヤルなど、回転させるタイプに向いている



# コネクタのプログラムを変えてみよう！（方向軸の変更）②

## X 軸の傾きを変更しよう

1 コネクタのプログラムを X 軸の傾きを取得するように変更します。スクーミーブロックエディタを起動させます。

▼スクーミー IDE      ▼スクーミーブロックエディタ      ▼回転軸：X

▼プログラム

2 「アップロード」をクリックし、スクーミーボードにプログラムを書き込みます。

アップロードする

3 シリアルモニタを表示させ「X」の値が表示されることを確認します。

スクーミーボードを上下に傾けて、「X」の値変化を確認する。  
 ボードが水平のとき 約 0 度  
 上に直角に傾けたとき 約 90 度  
 下に直角に傾けたとき 約 -90 度

▼上      ▼下

## Z 軸の傾きを変更しよう

4 スクーミーブロックエディタで、Z 軸の傾きを取得するように変更します。

▼プログラム      ▼回転軸：Z

5 「アップロード」をクリックし、スクーミーボードにプログラムを書き込みます。

アップロードする

6 シリアルモニタを表示させ「Z」の値が表示されることを確認します。

スクーミーボードを右左に回転させて、「Z」の値変化を確認する。  
 ボードが回転なしのとき 約 0 度  
 左に直角に回転させたとき 約 -90 度  
 右に直角に回転させたとき 約 90 度

▼左      ▼右

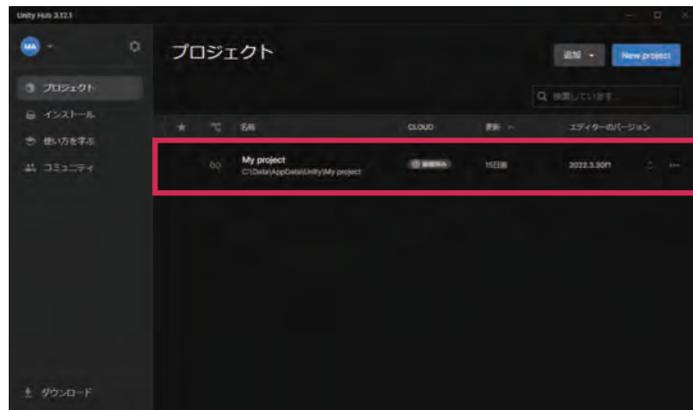
## 変更されたか確認しよう

7 動作確認をしましょう。Unity を起動させて、「あつめて！すいかばマン」のプロジェクトをクリックします。

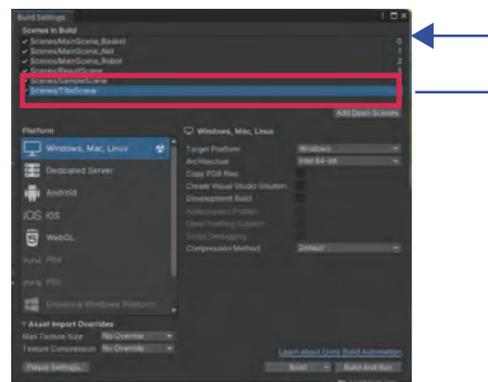
8 ゲームを実行させ、変更を確認します。Game 画面を開き、「▶」ボタンをクリックし、コネクタを動かし、方向軸が変わっているか確認しましょう。

## 作ったゲームを遊んでもらえるようにしよう！①

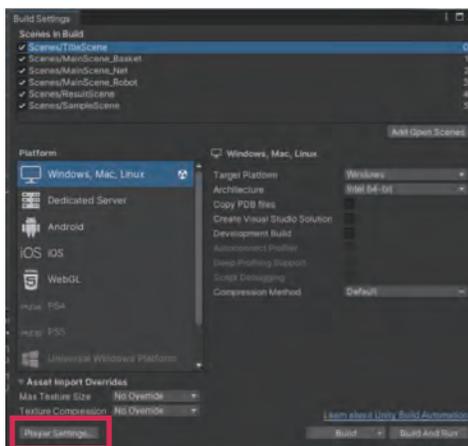
- 1 Unityを起動させて、「あつめて!すいかばマン」のプロジェクトをクリックします。



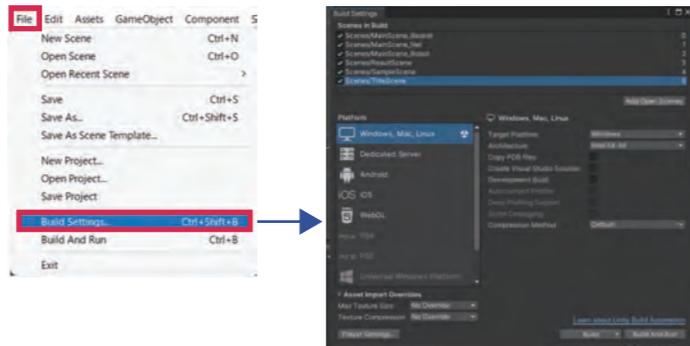
- 3 タイトル画面が初めに表示されるように、Scenesの順番を変更します。「TitleScene」を、ドラッグ&ドロップで一番上に移動させます。



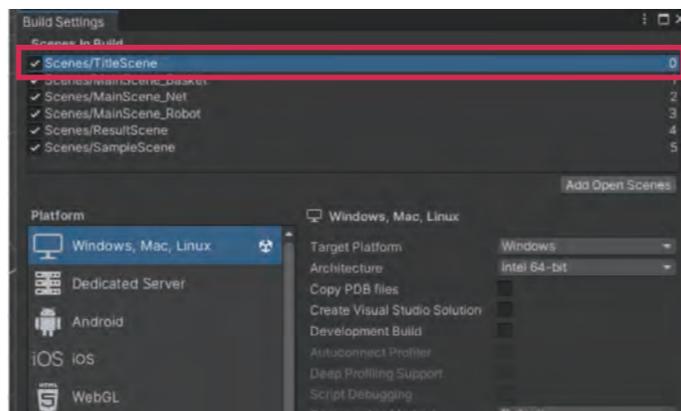
- 5 「Player Settings」ボタンをクリックします。



- 2 File → Build Settings をクリックします。

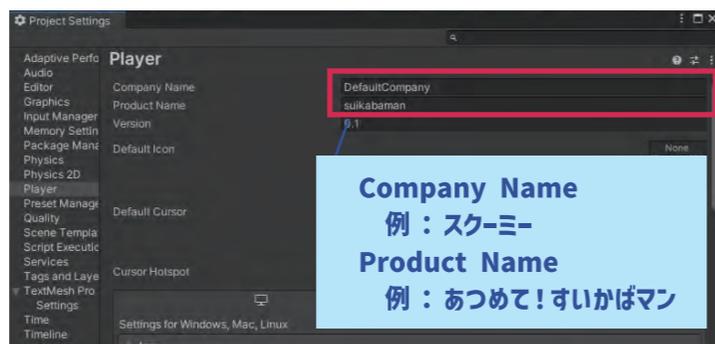


- 4 「TitleScene」が一番上になっていることを確認してください。

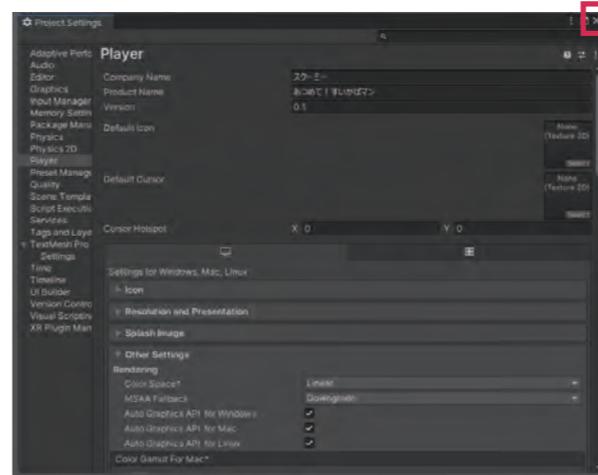


- 6 ゲーム情報を以下のように設定します。

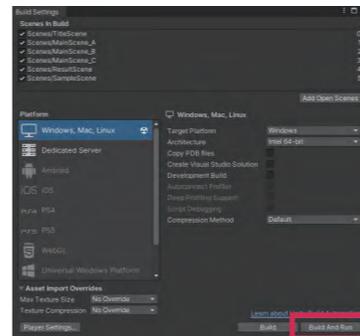
- Company Name (作成者名、所属、学校名等)
- Product Name (ゲームタイトル)



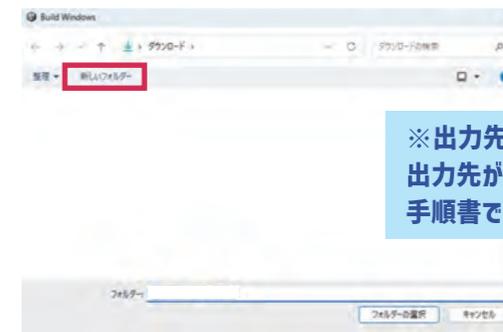
- 7 Player 設定画面の「×」ボタンをクリックして画面を閉じましょう。



- 9 実行ファイルを生成するため、「Build And Run」をクリックします。(動作確認を必要としない場合は「Build」をクリック)

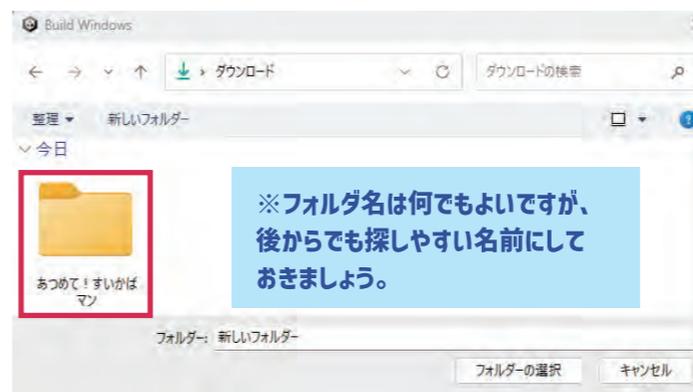


- 10 出力先のフォルダの作成をします。複数のファイルが作成されるので「新しいフォルダ」をクリックして、空のフォルダを作成します。



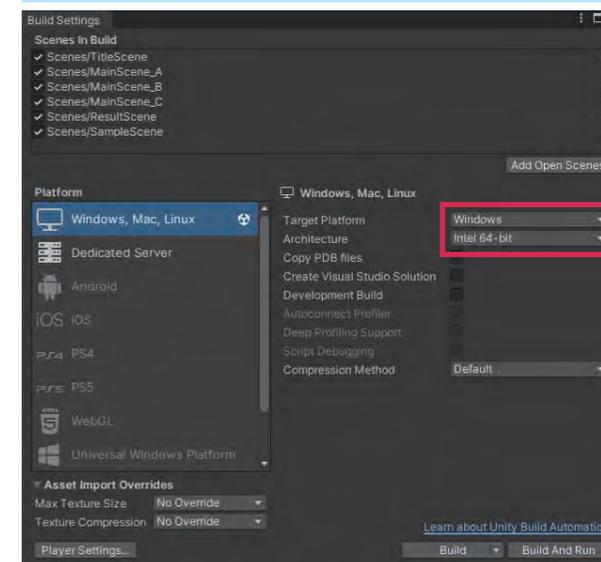
※出力先はどこでもよいですが、出力先がどこになるかは確認しておくこと。手順書では「ダウンロード」フォルダにしています。

- 11 フォルダの名前をゲーム名に変更して、フォルダをダブルクリックします。



- 8 Build Settings 画面で実行環境の設定をします。以下の設定を反映させましょう。

- Target Platform (windows、mac 等)
  - Architecture (Intel 64-bit、Intel 32-bit)
- ※実行する PC の環境に合わせてください。



- 12 フォルダの中身が空であることを確認して「フォルダ選択」をクリックします。



## 作ったゲームを遊んでもらえるようにしよう！②

- 13 ゲームの動作確認をしましょう。ゲーム画面が表示されるので「スタート」をクリックして、ゲームを開始します。



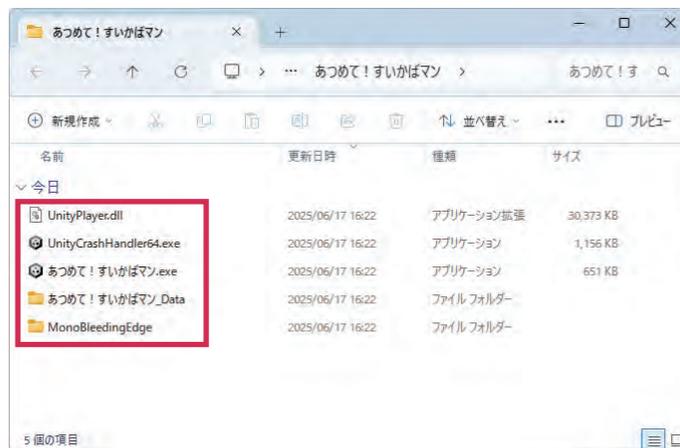
- 14 スクリーンボードを傾けてアイテムをキャッチできるか確認してください。



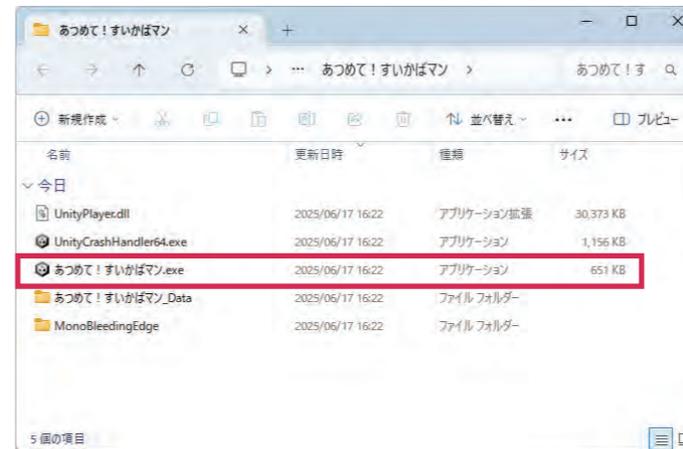
- 15 ゲーム終了の動作確認もしましょう。60秒待って、タイトル画面に戻り、「やめる」をクリックします。※60秒待たず、途中で終りたい場合は「Alt+F4キー」でゲーム終了してください。



- 16 配布用ゲーム資産の確認をします。エクスプローラーで出力先のフォルダを確認し、以下の5つのファイルが作成されていることを確認してください。



- 17 別のPCに資産をコピーして、配布用のゲームを動かしてみましょう。「あつめて! すいかばマン.exe」をダブルクリックします。(.exeはPCの設定によっては表示されません)



- 18 配布用ゲームの動作確認をしましょう。ゲーム画面が表示されるので「スタート」をクリックして、ゲームを開始します。



- 19 補足：キャラクターが動かなかった場合、SerialHandlerを修正してみる。「SerialHandler」を開き「portName」を配布先のPCの環境に合わせて修正後、配布データ作成（Build）を行いましょう。

